

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Yoshio HIROSE, et al.

Application No.:

Group Art Unit:

Filed: July 25, 2003

Examiner:

For: METHOD OF AND APPARATUS FOR INFORMATION PROCESSING

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2002-224074

Filed: July 31, 2002

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: July 25, 2003

By: 

H. J. Staas
Registration No. 22,010

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 7月31日

出 願 番 号

Application Number:

特願2002-224074

[ST.10/C]:

[JP 2002-224074]

出 願 人

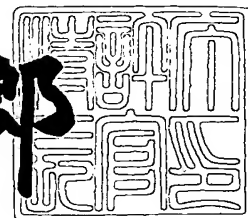
Applicant(s):

富士通株式会社

2003年 1月 7日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2002-3103561

【書類名】 特許願

【整理番号】 0240912

【提出日】 平成14年 7月31日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 7/00

【発明の名称】 情報処理装置

【請求項の数】 10

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 廣瀬 佳生

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 齋藤 美寿

【発明者】

 【住所又は居所】 オランダ王国 1851 エムピー ハイロー ブリーデラール 89

 【氏名】 ヴァウター クーゼイン

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100104190

 【弁理士】

 【氏名又は名称】 酒井 昭徳

【手数料の表示】

 【予納台帳番号】 041759

 【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9906241

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置

【特許請求の範囲】

【請求項 1】 与えられた命令が所定の命令セット内の命令であるか否かを判定する判定手段と、

前記判定手段により前記命令が所定の命令セット内の命令であると判定された場合に、当該命令を実行する第 1 の演算手段と、

前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令を実行するための回路構成を決定する構成情報を出力する構成情報出力手段と、

前記構成情報出力手段から出力された構成情報により決定される回路構成で前記命令を実行する第 2 の演算手段と、

を備えたことを特徴とする情報処理装置。

【請求項 2】 前記第 2 の演算手段を複数備えたことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】 前記構成情報出力手段は、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする請求項 1 または請求項 2 に記載の情報処理装置。

【請求項 4】 前記構成情報出力手段は、前記命令で指定されたアドレスおよび／または所定のレジスタに保持されたアドレスにもとづいて、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする請求項 3 に記載の情報処理装置。

【請求項 5】 前記構成情報は書き換え可能なメモリに保持されることを特徴とする請求項 1 ～ 4 のいずれか一つに記載の情報処理装置。

【請求項 6】 前記構成情報は前記メモリのほか、前記命令内の所定のフィールドあるいは所定のレジスタにも保持されることを特徴とする請求項 5 に記載の情報処理装置。

【請求項 7】 前記所定のレジスタは、前記第 2 の演算手段により前記命令

が実行される都度、前記メモリに保持された構成情報にもとづいて値を更新することを特徴とする請求項 6 に記載の情報処理装置。

【請求項 8】 さらに、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定する第 2 の判定手段と、

前記第 2 の判定手段により、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリに前記構成情報をロードする命令を複数発行する命令発行手段と、を備え、

前記第 1 の演算手段は、前記命令発行手段により発行された命令を実行することを特徴とする請求項 5 ～ 7 のいずれか一つに記載の情報処理装置。

【請求項 9】 さらに、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定する第 2 の判定手段と、

前記第 2 の判定手段により、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリが割り当てられた所定のレジスタに前記構成情報を転送する命令を発行する命令発行手段と、を備え、

前記第 1 の演算手段は、前記命令発行手段により発行された命令を実行することを特徴とする請求項 5 ～ 7 のいずれか一つに記載の情報処理装置。

【請求項 10】 さらに、前記構成情報出力手段により出力された構成情報中所定のビット位置にある情報のみを出力する選択手段を備えたことを特徴とする請求項 1 ～ 9 のいずれか一つに記載の情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、パーソナルコンピュータや各種の通信機器、AV機器、あるいは家電などの情報処理装置、より詳細には当該装置に搭載されるプロセッサに関する。

【0002】

【従来の技術】

半導体の微細加工技術の進展とともに、L S I チップに搭載される回路規模が大きくなり、プロセッサ（C P U）とともにユーザ独自の回路を1チップに集積したシステムL S Iの開発が従来から行われている。

【0 0 0 3】

そして、ユーザのアプリケーションに適したシステムL S Iを開発する方式として、（1）プロセッサとは独立にアプリケーションに応じた回路ブロックを搭載する方法と、（2）プロセッサ自体をアプリケーションに応じて修正する方法とがある。

【0 0 0 4】

このうち、（1）はごく一般のシステムL S Iの実現方法である。（2）はT e n s i l c a社やA R C社のコンフィギャラブルプロセッサに見られる手法であり、ユーザのアプリケーションに適した専用命令（カスタム命令）を命令セットに追加するとともに、当該命令を実行するための演算器をプロセッサに追加するというものである。

【0 0 0 5】

【発明が解決しようとする課題】

しかしながらプロセスの微細化に伴って、マスク代などのプロセスコストは非常に高額となる。そしてその費用にペイするのは、非常に大量に生産される一部のシステムL S Iに限られてしまい、そうでないものはプロセス費用がチップ単価に大きく跳ね返る結果、経済的に引き合わないものになってしまう。すなわち、上記（1）（2）は大量生産されるシステムL S Iについては有効であるが、少量しか必要とされないシステムL S Iにはコスト的に適用できない。

【0 0 0 6】

上記に対し近年になって、（3）プロセッサとF P G Aとをチップ上に混載する方法が現れ、実際に製品化されてきている。この手法によると、ユーザのアプリケーションに応じた様々な回路をF P G Aブロックで実現できるが、F P G Aの面積効率は通常の回路に対して1 / 1 0 ~ 1 / 2 0と低く、やはりコスト高になってしまうという問題点があった。さらにF P G A部分は動作周波数が高くないので、性能を引き上げるためには並列化する必要がある、これに伴って回

路規模が大きくなってしまいう問題点があった。

【0007】

この発明は上記従来技術による問題を解決するため、個々に回路を作り替えたり大規模な回路を搭載したりしなくても、ユーザが定義する様々な命令を高速に処理することが可能な情報処理装置を提供することを目的とする。

【0008】

【課題を解決するための手段】

上述した課題を解決し、目的を達成するため、この発明にかかる情報処理装置は与えられた命令が所定の命令セット内の命令であるか否かを判定し、そうであると判定された場合には第1演算器（従来型の演算器）により当該命令を実行するとともに、そうでないと判定された場合には、第2演算器（本発明による演算器）に当該命令を実行するための回路構成を決定する構成情報を出力し、当該回路構成となった第2演算器により当該命令を実行することを特徴とする。

【0009】

また、この発明にかかる情報処理装置は、第2演算器を複数備えたことを特徴とする。

【0010】

また、この発明にかかる情報処理装置は、前記命令が所定の命令セット内の命令でないと判定された場合に、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする。

【0011】

また、この発明にかかる情報処理装置は、前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令で指定されたアドレスおよび／または所定のレジスタに保持されたアドレスにもとづいて、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする。

【0012】

また、この発明にかかる情報処理装置は、前記構成情報が書き換え可能なメモリに保持されることを特徴とする。

【 0 0 1 3 】

また、この発明にかかる情報処理装置は、前記構成情報がメモリのほか命令内の所定のフィールドあるいは所定のレジスタにも保持されることを特徴とする。

【 0 0 1 4 】

また、この発明にかかる情報処理装置は、前記第 2 の演算器により命令が実行される都度、前記メモリに保持された構成情報にもとづいて、前記レジスタが値を更新することを特徴とする。

【 0 0 1 5 】

また、この発明にかかる情報処理装置は、前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定し、そうであった場合には前記メモリに前記構成情報をロードする命令を複数発行して、第 1 演算器で各々の命令を実行することを特徴とする。

【 0 0 1 6 】

また、この発明にかかる情報処理装置は、前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定し、そうであった場合には前記メモリが割り当てられた所定のレジスタに前記構成情報を転送する命令を発行して、第 1 演算器で当該命令を実行することを特徴とする。

【 0 0 1 7 】

また、この発明にかかる情報処理装置は、前記構成情報中所定のビット位置にある情報のみを出力することを特徴とする。

【 0 0 1 8 】

これらの発明により実現されるプロセッサを搭載した情報処理装置は、与えられた命令が命令セット内に用意されていないような命令（たとえば入力データ中のビットを入れ替える処理や、1 の個数を係数する処理など）であっても、メモリから読み出した構成情報により自らの演算器の構成を変化させることで、高速に当該命令を実行できる。逆に言えば、適切な構成情報さえ用意してやれば、上記装置は原理上は無数の種類の命令を処理することができる。

【 0 0 1 9 】

【発明の実施の形態】

以下に添付図面を参照して、この発明にかかる情報処理装置の好適な実施の形態を詳細に説明する。

【 0 0 2 0 】

(本発明の基本構成)

具体的な実施例の説明に入る前に、本発明の基本構成について説明する。図 1 は、従来技術によるプロセッサの内部構成を模式的に示すブロック図である。命令キャッシュ 1 0 0 に蓄積された命令は、命令フェッチ部 1 0 1 により読み出され、命令デコーダ 1 0 2 で解釈されて、その解釈に応じてレジスタファイル 1 0 3 や演算器 1 0 4 に信号が送られる。

【 0 0 2 1 】

この演算器 1 0 4 は、プロセッサの設計時にあらかじめ定義された命令セット内の命令しか実行できない。そのため、一般的なプロセッサの命令セットには用意されていないような複雑な処理、たとえばデータ中のビットの入れ替えなどを行うには、命令セット内の命令をいくつか組み合わせてやる必要がある。そして、組み合わせる命令の個数が多いほど全体の処理時間も長くなってしまふ。

【 0 0 2 2 】

もっとも逆に言えば、あらかじめ想定されていない処理でも 1 つのカスタム命令として定義し、当該命令を実行するような演算器をプロセッサに組み込むことができれば、処理を高速化することができる。

【 0 0 2 3 】

図 2 は、本発明によるプロセッサの内部構成を模式的に示す説明図である。図示するように本発明によるプロセッサは、図 1 に示した従来技術のプロセッサに、その内部構成を外部からの構成情報の入力により変化させることができる第 2 演算器 2 0 5、および当該構成情報を蓄積・供給するコンフィギュレーションメモリ 2 0 6 を追加した構成である。

【 0 0 2 4 】

第 2 演算器 2 0 5 は、具体的には複数の AND 回路、OR 回路、加算器、セレ

クタ、マルチプレクサなどから構成されている。なお、図示する例では一個のみであるが、プロセッサ内に第2演算器205を複数設ける構成にしてもよい。この場合、演算器ごとにコンフィギュレーションメモリ206を用意してもよいが、複数の演算器でメモリを共有する、すなわち読み出しポートを第2演算器205の数だけ持たせるようにして、メモリの節約をはかることもできる。

【0025】

また、図3はコンフィギュレーションメモリ206の内容を模式的に示す説明図である。図示するように、コンフィギュレーションメモリ206には第2演算器205に供給するための複数の構成情報が蓄積されている。

【0026】

この構成情報とは具体的には、どのカスタム命令が入力したときは第2演算器205内のどのマルチプレクサで何番目のビットを選択すればよいか、などといったパラメータを定義したものである。1つのエントリに格納された構成情報が1つのカスタム命令に対応し、構成情報の読み出し元のアドレスを変化させる、すなわち供給する構成情報を変化させることで、第2演算器205に個々のカスタム命令に応じた処理を行わせることができる。

【0027】

(出力する構成情報の切り替え処理)

次に、出力する構成情報の切り替え処理について説明する。第2演算器205に供給する構成情報を切り替えるには、たとえば下記のようにして当該情報の格納されたアドレスを指定してやればよい。

【0028】

(1) 命令コードによる指定

図4に示すように、命令コード中の何ビットかをコンフィギュレーションメモリ206のアドレス指定のために割り当てておけば、当該メモリ内のどの構成情報を第2演算器205に出力するか、すなわち第2演算器205にどのカスタム命令を実行させるかを指定できる。

【0029】

なお、カスタム命令ごとに異なるオペコードを割り当ててもよいが、そうする

と定義できるカスタム命令の数がオペコードの種類の上限に制限されてしまうので、ここではカスタム命令であればオペコードは共通（１種類だけ）で、それに続くアドレスで個々のカスタム命令を区別している。

【0030】

（２）特定用途レジスタによる指定

上記のように命令コード中でメモリアドレスを指定すると、標準的な３２ビット長の命令の場合、アドレスの指定用にはせいぜい５ビット程度しか取れないため、指定できるエントリは３２個程度に限定されてしまう。

【0031】

そこで別の方法として、コンフィギュレーションメモリ２０６のアドレスを保持する特定用途レジスタ（ＳＰＲ）を用意し、当該レジスタに、カスタム命令の実行に先立って当該命令に対応する構成情報のアドレスをセットするようにしてもよい。特定用途レジスタは通常は３２ビットなので、命令コード中の数ビットを使うよりも多くのアドレスを指定できる。逆に言えば、より大規模なコンフィギュレーションメモリ２０６を搭載することが可能である。

【0032】

もっとも特定用途レジスタを利用する場合は、コンフィギュレーションメモリ２０６のアドレスを当該レジスタにセットするステップと、当該アドレスの構成情報を第２演算器２０５に供給して、命令を実際に実行するステップとの２段階でカスタム命令が実行されることになる。ただし同一のカスタム命令を続けて実行する場合は、特定用途レジスタへのアドレス設定は毎回行う必要はないので、このオーバーヘッドは発生しない。

【0033】

（３）命令コードおよび特定用途レジスタによる指定

上記（１）および（２）を組み合わせ、メモリアドレスの一部（たとえば上位アドレス）を特定用途レジスタで、残りの一部（たとえば下位アドレス）を命令コードで指定すれば、欠点を相互に補い合い、大規模なメモリを搭載しつつ効率的にカスタム命令を実行してゆくことができる。

【0034】

(メモリ内の構成情報の切り替え処理)

次に、メモリ内の構成情報の切り替え処理について説明する。コンフィギュレーションメモリ 2 0 6 として ROM のような読み出し専用メモリを使うと、第 2 演算器 2 0 5 で実行可能なカスタム命令の種類は、上記メモリのエントリの個数に制限される。

【 0 0 3 5 】

一方、コンフィギュレーションメモリ 2 0 6 として RAM を使うと、プログラム実行中にメモリ内の情報を書き換えることで、エントリの個数以上の（理論上は無限の）カスタム命令を実行させることができる。コンフィギュレーションメモリ 2 0 6 への構成情報のロードは、たとえば下記のようにして実現すればよい。

【 0 0 3 6 】

(1) 実現方法 1 : 命令セットにコンフィギュレーションメモリ 2 0 6 へのロード命令を追加する。

命令セット中に、コンフィギュレーションメモリ 2 0 6 へのロード命令を新たに追加し、当該命令を実行できるように演算器 2 0 4 を拡張する。演算器 2 0 4 は、上記ロード命令を命令デコーダ 2 0 2 から供給されると、当該命令で指定されたアドレスの構成情報をコンフィギュレーションメモリ 2 0 6 の指定されたエントリにロードする。

【 0 0 3 7 】

(2) 実現方法 2 : コンフィギュレーションメモリ 2 0 6 へのロード命令を複数のロード命令で代替する。

あるいは命令フェッチ部 2 0 1 がコンフィギュレーションロード命令を検出すると、命令デコーダ 2 0 2 には上記命令の代わりに、必要な数のロード命令を供給するようにし、ロードされたデータをコンフィギュレーションメモリ 2 0 6 に格納するようにしてもよい。たとえばロードする構成情報が 2 5 6 ビットの場合は、コンフィギュレーションロード命令の代わりに、8 個のロード命令あるいは 4 個のダブルワードロード命令を命令デコーダ 2 0 2 に供給する。

【 0 0 3 8 】

(3) 実現方法3：コンフィギュレーションメモリ206を特定用途レジスタにマッピングする。

命令セットに含まれている、特定用途レジスタへのデータの転送命令を利用してもよい。すなわち、コンフィギュレーションメモリ206を特定用途レジスタにマッピングし、汎用レジスタへのロード命令と、当該汎用レジスタから上記特定用途レジスタへの転送命令とを実行することでコンフィギュレーションメモリ206へ構成情報をロードする。プログラム中でこれらの命令を組み合わせることでコンフィギュレーションメモリ206へのロードを実現できるので、あえて命令セット中にコンフィギュレーションロード命令を用意する必要はない。ただ、プログラムの負担を軽減する意味では命令セットに上記命令を追加し、命令デコーダ202で当該命令を汎用レジスタへのロード命令と、特定用途レジスタへの転送命令とに置換するようにしてもよい。

【0039】

(4) 実現方法4：コンフィギュレーションメモリ206を通常のアドレス空間にマッピングする。

コンフィギュレーションメモリ206を通常のアドレス空間にマッピングし、レジスタへのロード命令に続けて上記メモリが割り当てられた領域へのストア命令を実行するか、DMAコントローラを起動して、外部メモリからコンフィギュレーションメモリ206へ直接データを転送させることで構成情報をロードすることもできる。プログラム中でロード命令とストア命令とを組み合わせることで、あるいはDMAを起動することで、コンフィギュレーションメモリ206へのロードを実現できるので、あえて命令セット中にコンフィギュレーションロード命令を用意する必要はない。ただ、プログラムの負担を軽減する意味では命令セットに上記命令を追加し、命令デコーダ202で当該命令をロード命令とストア命令、あるいはDMAの起動命令に置換するようにしてもよい。

【0040】

(実施の形態1)

次に、実施の形態1、すなわち、ビットの入れ替え処理について説明する。図5は本発明の実施の形態1による第2演算器205の内部構成を模式的に示す説

明図である。第2演算器205内の32個のマルチプレクサ（同図には3個しか示していないが実際には32個）に、それぞれ異なる位置のビットを出力するよう選択信号を与えてやると、レジスタファイル203からの入力データ（32ビットであるものとする）の任意のビットを、出力データの任意のビットに持ってゆくことができる。すなわち、あらゆるパターンのビットシャッフリングが可能である。

【0041】

このとき、32ビット中の1ビットを指定するには少なくとも5ビットが必要なので、コンフィギュレーションメモリ206から第2演算器205に供給される選択信号、すなわち構成情報は最低でも5ビット×32個＝160ビットとなる。

【0042】

またレジスタファイル203からの入力データとして、1と0とを少なくとも1つ含むデータ、すなわちオール0（「000・・・000」）やオール1（「111・・・111」）以外のビット列を与えてやると、選択信号を適当に作ってやれば、あらゆるビットパターンの出力データを生成することができる。すなわち、任意のマスクパターンを出力するパターンジェネレータが実現できる。

【0043】

上記の入力データとしては、たとえばLSBのみに1が立っているデータを使用する。このような入力データは汎用レジスタに、ユーザがプログラムで明示的に設定しておくこともできるが、汎用レジスタのいずれかをこのデータ専用割り当てしておくこともできる。たとえば、一般的なプロセッサの0番レジスタ（GR0）には固定値0が設定されているが、同様にたとえば1番レジスタ（GR1）には固定値として1（LSBのみに1が立っているデータ）を設定しておけば、わざわざ値を設定してやる必要がない。

【0044】

（実施の形態2）

次に、実施の形態2、すなわち、1が立っているビットの計数処理について説明する。図6は本発明の実施の形態2による第2演算器205の内部構成を模式

的に示す説明図である。実施の形態 2 では、第 2 演算器 2 0 5 内の右シフタ 6 0 0、パターンジェネレータ 6 0 1、AND 回路 6 0 2・6 0 3 および加算器 (ALU でもよい) 6 0 4 を用いて、図 7 に示す処理を所定回数だけ繰り返すことで、入力データ中で 1 が立っているビットの個数をカウントする。

【0 0 4 5】

説明の便宜上、ここでは入力データとして 8 ビットのデータ、たとえば「a b c d e f g h」を考える。ここで「a」「b」などは、そのビット位置にある 0 または 1 を表すものとする。

【0 0 4 6】

(1) 1 回目の処理

図 8 において、まず、右シフタ 6 0 0 により入力データを右に所定ビット (1 回目では 1 ビット) だけシフトする (ステップ S 7 0 1)。次にパターンジェネレータ 6 0 1 により、所定のマスクパターン (1 回目では「0 1 0 1 0 1 0 1」すなわち 0 x 5 5) を生成する (ステップ S 7 0 2)。さらに、一方の AND 回路 6 0 2 で入力データと上記マスクパターンとの論理積を (ステップ S 7 0 3)、もう一方の AND 回路 6 0 3 で右シフト後の入力データと上記マスクパターンとの論理積をそれぞれ取った後 (ステップ S 7 0 4)、これら 2 つの AND 結果を加算器 6 0 4 で加算する (ステップ S 7 0 5)。

【0 0 4 7】

上記処理の結果、加算器 6 0 4 からの出力データでは、図 8 に示すようにその最上位の 2 ビットが入力データ中の a と b との加算結果、次の 2 ビットが c と d との加算結果、次の 2 ビットが e と f との加算結果、最下位の 2 ビットが g と h との加算結果となっている (偶数ビット (2 n) と奇数ビット (2 n + 1) の和が計算されたと言ってもよい)。そして、ステップ S 7 0 1 ~ S 7 0 5 の繰り返し回数が所定の回数 (入力データが 8 ビットの場合は 3 回) に達しない間は (ステップ S 7 0 6 : No)、再度ステップ S 7 0 1 に移行する。

【0 0 4 8】

(2) 2 回目の処理

図 9 において、2 回目の処理では、その直前の 1 回目の処理で得られた出力デ

ータが入力データとなる。まず、右シフタ600により上記入力データを右に所定ビット（2回目では2ビット）だけシフトする（ステップS701）。次にパターンジェネレータ601により、所定のマスクパターン（2回目では「00110011」すなわち0x33）を生成する（ステップS702）。そして、これらの入力データとマスクパターンとの間で、1回目と同様論理積とその和とを計算する（ステップS703～S705）。

【0049】

上記処理の結果、加算器604からの出力データでは、図9に示すようにその上位4ビットが入力データ中の $a \cdot b \cdot c \cdot d$ の加算結果、下位4ビットが $e \cdot f \cdot g \cdot h$ の加算結果となっている。そして、ステップS701～S705の繰り返し回数がまだ所定回数（3回）に達しないので（ステップS706：No）、再びステップS701に戻る。

【0050】

（3）3回目の処理

図10において、3回目の処理では、その直前の2回目の処理で得られた出力データが入力データとなる。まず、右シフタ600により上記入力データを右に所定ビット（3回目では4ビット）だけシフトする（ステップS701）。次にパターンジェネレータ601により、所定のマスクパターン（3回目では「00001111」すなわち0x0F）を生成する（ステップS702）。そして、これらの入力データとマスクパターンとの間で、1回目や2回目と同様論理積とその和とを計算する（ステップS703～S705）。

【0051】

上記処理の結果、加算器604からの出力データには、図10に示すように入力データ中の各ビットの加算結果、すなわち入力データ中で1が立っているビットの個数が格納されている。なお、上記データの最下位ビットは入力データのパリティビットにもなっている。そして、この時点でステップS701～S705の繰り返し回数が所定回数（3回）に達したので（ステップS706：Yes）、上記フローチャートによる処理を終了する。

【0052】

なお、この繰り返しの回数は入力データのビット数に依存し、8ビットの場合は上述のように3サイクル、32ビットの場合は5サイクルである。ビットの増加に伴って繰り返しの回数が増え、右シフタ600によるシフト量やパターンジェネレータ601により生成されるマスクパターンも変化してゆくが、図7に示した処理の手順は同一である。図11に、入力データが8ビットである場合と32ビットである場合のシフト量およびマスクパターンをそれぞれ示す。

【0053】

また、パターンジェネレータ601の生成パターンを変えることによって、上記のような1のカウント処理以外にも様々な処理を実現できる。また、上記ではマスクパターンをパターンジェネレータ601で生成しているが、コンフィギュレーションメモリ206にマスクパターンを格納しておいて、それを直接使うようにしてもよい。

【0054】

(パラメータの指定方法)

次に、パラメータの指定方法について説明する。上述のように各回の処理ごとに異なるのは、右シフタ600でのシフト量とパターンジェネレータ601で生成されるマスクパターンのみである。そこで、コンフィギュレーションメモリ206内に各回ごとの構成情報を用意し（各回ごとに異なるカスタム命令を用意し、と言ってもよい）、これらを順次第2演算器205に与えることで上記一連の処理を実現してもよいが、一部のパラメータが異なるだけの複数の構成情報でメモリを占有するのは無駄である。

【0055】

そこで、各回ともコンフィギュレーションメモリ206から読み出す構成情報は共通とし（命令中で指定する上記メモリのアドレスは同一とし、と言ってもよい）、ただ各回ごとに異なるパラメータのみを、以下に例示するように所定のレジスタにロードしておいたり、命令中に当該パラメータ用のフィールドを設けたりすることで、別途指定するようにしてもよい。第2演算器205の回路構成を決定するためのパラメータが、コンフィギュレーションメモリ206のほか、所定のレジスタや命令中に分散していると見ることもできる。

【 0 0 5 6 】

(1) 特定用途レジスタによる指定

右シフタ 6 0 0 用のパラメータ指定用、パターンジェネレータ 6 0 1 用のパラメータ指定用の特定用途レジスタをそれぞれ用意しておき、構成情報中では当該レジスタを指定する。この場合、右シフタ 6 0 0 用の特定用途レジスタにシフト量をセット→パターンジェネレータ 6 0 1 用の特定用途レジスタにマスクパターン生成のためのパラメータをセット→カスタム命令を実行、という 3 ステップの処理を 3 セット繰り返すことで、上述のカウント処理を実現できる。

【 0 0 5 7 】

(2) 自動更新機能を有する特定用途レジスタによる指定

上記 (1) では、構成情報を保持するためのメモリは少なくて済む一方、3 ステップ×3 セット=9 ステップの処理が必要になってしまう。メモリを節約しながら処理時間を短縮するには、たとえば図 1 2 に示すような回路で、右シフタ 6 0 0 用・パターンジェネレータ 6 0 1 用の特定用途レジスタに自動更新機能を具備することが考えられる。

【 0 0 5 8 】

図示する例ではレジスタ内の現在の値と、コンフィギュレーションメモリ 2 0 6 内の値との和を更新後の値としている。もっとも加算値が固定、たとえば 1 の場合は、特にコンフィギュレーションメモリ 2 0 6 から当該値を供給してやる必要はない。なお、図示する例ではこの (2) の方式と、後述する (3) の方式のいずれか一方をセレクタで選択できるようになっている。どちらを選択するかはコンフィギュレーションメモリ 2 0 6 内の構成情報により決定される。

【 0 0 5 9 】

この (2) の方式では、特定用途レジスタへの設定は最初の 1 回だけ行えばよい (2 回目以降は自動更新される) ので、必要なステップを (1) の 9 ステップから 5 ステップに減らすことができる。

【 0 0 6 0 】

(3) 命令コードによる指定

別の方法として、メモリを節約しながら処理時間を短縮するには、たとえば命

令コード中で右シフタ 6 0 0 やパターンジェネレータ 6 0 1 のパラメータを直接指定することも考えられる。これにより、毎回特定用途レジスタを設定する必要がなくなるので、同じカスタム命令（ただしパラメータはそれぞれ異なる）を 3 回実行するだけで上述のカウント処理を実現できる。もっとも、命令長は通常 3 2 ビットに制限されることから、パラメータの指定のために 1 0 数ビット確保するのは難しい場合が多い。

【 0 0 6 1 】

(4) 汎用レジスタによる指定

実施の形態 2 で扱う入力データは 1 つなので、命令のフォーマットが図 4 のようなものであった場合、もう 1 つソースレジスタを指定することができる。ここで指定される汎用レジスタの特定フィールドに、右シフタ 6 0 0 やパターンジェネレータ 6 0 1 に与えるパラメータをロードしておくこともできる。

【 0 0 6 2 】

(実施の形態 3)

次に、DES の IP 転置処理について説明する。図 1 3 は本発明の実施の形態 3 による第 2 演算器 2 0 5 の内部構成を模式的に示す説明図である。実施の形態 3 では、第 2 演算器 2 0 5 内の右シフタ 1 3 0 0、パターンジェネレータ 1 3 0 1、反転回路 1 3 0 2、AND 回路 1 3 0 3・1 3 0 4、OR 回路 1 3 0 5 およびビット入れ替え器 1 3 0 6 を用いて、図 1 4 に示す手順により、6 4 ビットの入力データについて DES の IP 転置処理を行う。

【 0 0 6 3 】

暗号化アルゴリズムの DES ではビットの入れ替え操作を多用するが、その中の IP 転置操作では図 1 5 のような 6 4 ビットのデータに対して、図 1 6 に示すようなビット入れ替えを行う。なお、各数字はビット位置を示している。

【 0 0 6 4 】

図 1 5 の転置前の入力データは、汎用レジスタ上では 2 つのレジスタに分割されて、たとえば 3 0 番 (GR 3 0) に 1 ~ 3 2 ビット目まで (1, 2, ..., 3 2) が、3 1 番 (GR 3 1) に 3 3 ~ 6 4 ビット目まで (3 3, 3 4, ..., 6 4) が、それぞれ格納される。また、図 1 6 の転置後の出力データは前半の

32ビット(58, 50, ..., 8)が28番(GR28)、後半の32ビット(57, 49, ..., 7)が29番(GR29)というように、2分割されて汎用レジスタに格納される。

【0065】

図13に示した第2演算器205では、コンフィギュレーションメモリ206からの構成情報(あるいは特定用途レジスタからのパラメータ、など)により、右シフタ1300はデータを4ビットだけ右シフトするように、パターンジェネレータ1301はマスクパターン0xFF0F0F0Fを出力するように、それぞれ設定されている。

【0066】

また、ビット入れ替え器1306は図17に示すような規則で各ビットの順序を入れ替える。図中「*」は任意のビット位置のデータでよいことを示し、たとえば最上位ビットの値を一律に埋めるようにする。

【0067】

上記を前提として、図14に示した手順を具体的に説明する。まず、第2演算器205で図18に示すように、入力1側にGR0(一般のプロセッサでよく使われるゼロレジスタで、読み出し値が常にオール0になる特殊なレジスタ)、入力2側にGR30を選択すると、出力レジスタ(ここではGR28とする)には図16に示した転置後のデータで、右上隅の4×4ビットにあたる部分が左詰めで出力される(ステップS1401)。

【0068】

次に、図19に示すように、入力1側にステップS1401による処理後のGR28、入力2側にGR31を選択すると、出力レジスタ(ここではGR28とする)には図16に示した転置後のデータで、上半分の8×4ビットにあたる部分が出力される(ステップS1402)。

【0069】

次に、演算器204に通常の命令(すなわち命令セット内の命令)である右シフト命令を与えて、GR30のデータを右に1ビットシフトし、結果をGR30に出力する(ステップS1403)。また、同様に右シフト命令を使って、GR

31のデータを右に1ビットシフトし、結果をGR31に出力する（ステップS1404）。

【0070】

次に、図20に示すように、入力1側にGR0、入力2側にステップS1403による処理後のGR30を選択すると、出力レジスタ（ここではGR29とする）には図16に示した転置後のデータで、右下隅の4×4ビットにあたる部分が左詰めで出力される（ステップS1405）。

【0071】

次に、図21に示すように、入力1側にステップS1405による処理後のGR29を、入力2側にステップS1404による処理後のGR31を選択すると、出力レジスタ（ここではGR29とする）には図16に示した転置後のデータで、下半分の8×4ビットにあたる部分が出力される（ステップS1406）。

【0072】

以上により、GR30およびGR31に分割して格納されていた64ビットの入力データが、ビット入れ替えの上でGR28およびGR29に出力されたことになり、図15から図16へのIP転置処理が終了する。従来のプロセッサでは1ビットずつデータを操作していたので、全体の入れ替えに多くの命令を実行する必要があったが、上述した実施の形態3ではIP転置処理を6命令で実現することができ、実行速度の大幅な向上がはかれる。

【0073】

なお、64ビットデータを一挙に扱うことのできるビット入れ替え器を用いれば、1命令で上記処理を実現できるが、その分回路規模が大きくなることは避けられない。

【0074】

以上説明した実施の形態1～3によれば、ユーザが求める種々のカスタム命令をコンフィギュレーションメモリ206からの構成情報を切り替えるだけで、すなわちLSIを個々に作り替えたり大規模なFPGAを搭載したりしなくても定義することができるので、ユーザのアプリケーションに適したカスタムプロセッサを非常に容易、かつ安価に実現できる。

【 0 0 7 5 】

また、従来の演算器 2 0 4 なら多数の命令の組み合わせで実現しなければならなかった処理を、第 2 演算器 2 0 5 ではごく少数の命令で実現できるので、プロセッサそのものの処理速度の向上だけでなく、当該プロセッサで実行されるプログラムのサイズの縮小も同時に実現することができる。

【 0 0 7 6 】

なお、上述した実施の形態 1 ～ 3 の第 2 演算器 2 0 5 は、コンフィギュレーションメモリ 2 0 6 からの情報をもっぱら自己の内部構成を決定するために用いたが、コンフィギュレーションメモリ 2 0 6 をダイレクトにテーブルメモリとして扱い、第 2 演算器 2 0 5 はメモリ内の任意の情報を読み出すだけにすることもできる。

【 0 0 7 7 】

すなわち、コンフィギュレーションメモリ 2 0 6 の 1 エントリ分の情報が 2 5 6 ビットだとすると、これを 1 ビット × 2 5 6、2 ビット × 1 2 8、4 ビット × 6 4 のようなテーブルと見なし、与えられた命令の特定のフィールドを当該テーブルのアドレスと見なして、そのアドレスのデータを読み出し出力する。

【 0 0 7 8 】

この場合の第 2 演算器 2 0 5 は、実施の形態 1 ～ 3 で上述した種々のカスタム命令を処理する演算器部分（コンフィギュレーションメモリ 2 0 6 からの構成情報によりその内部構成が変化する部分、と言ってもよい）と、コンフィギュレーションメモリ 2 0 6 内の任意のデータを抽出するための選択器部分とからなる。

【 0 0 7 9 】

図 2 2 は、この選択器部分の回路構成を示す説明図である。図示する例では、コンフィギュレーションメモリ 2 0 6 から入力した 2 5 6 ビットのデータから、3 2 対 1 の 8 個のマルチプレクサにより 8 ビットのデータのみが選択（抽出）されている。なお、個々のマルチプレクサが必要とする選択信号（3 2 ビット中のどのビットを選択すればよいかを示す信号であり、長さは 5 ビットである）は、汎用レジスタや特定用途レジスタ、あるいは命令内の特定のフィールドなどから供給する。

【 0 0 8 0 】

上記はコンフィギュレーションメモリ 2 0 6 を 8 ビット×3 2 のテーブルとして使用する例であるが、4 ビット×6 4 のテーブルとして使用するのであれば、図 2 3 に示すように選択信号を 1 ビット増やして、3 2 対 1 の 8 個のマルチプレクサから出力された 8 ビットを、さらに 2 対 1 の 4 個のマルチプレクサで 4 ビットに絞ればよい。

【 0 0 8 1 】

また、コンフィギュレーションメモリ 2 0 6 を 2 ビット×1 2 8 のテーブルとして使用するのであれば、さらに選択信号を 1 ビット増やし、2 対 1 の 2 個のマルチプレクサで 2 ビットに絞ればよい。もう一段 2 対 1 のマルチプレクサを通せば最終的に 1 ビットの出力となり、1 ビット×2 5 6 のテーブルとして使うこともできる。

【 0 0 8 2 】

なお、選択器の構成を図 2 4 のようにすると出力データの幅が 2 のべき乗に制限されるため、テーブルの作り方は複雑になる一方、図 2 3 よりも回路量を削減しつつ同一の機能を実現することができる。

【 0 0 8 3 】

（付記 1）与えられた命令が所定の命令セット内の命令であるか否かを判定する判定手段と、

前記判定手段により前記命令が所定の命令セット内の命令であると判定された場合に、当該命令を実行する第 1 の演算手段と、

前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令を実行するための回路構成を決定する構成情報を出力する構成情報出力手段と、

前記構成情報出力手段から出力された構成情報により決定される回路構成で前記命令を実行する第 2 の演算手段と、

を備えたことを特徴とする情報処理装置。

【 0 0 8 4 】

（付記 2）前記第 2 の演算手段を複数備えたことを特徴とする付記 1 に記載の情

報処理装置。

【 0 0 8 5 】

（付記 3）前記構成情報出力手段は、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする付記 1 または 2 に記載の情報処理装置。

【 0 0 8 6 】

（付記 4）前記構成情報出力手段は、前記命令で指定されたアドレスおよび／または所定のレジスタに保持されたアドレスにもとづいて、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする付記 3 に記載の情報処理装置。

【 0 0 8 7 】

（付記 5）前記構成情報は書き換え可能なメモリに保持されることを特徴とする付記 1 ～ 4 のいずれか一つに記載の情報処理装置。

【 0 0 8 8 】

（付記 6）前記構成情報は前記メモリのほか、前記命令内の所定のフィールドあるいは所定のレジスタにも保持されることを特徴とする付記 5 に記載の情報処理装置。

【 0 0 8 9 】

（付記 7）前記所定のレジスタは、前記第 2 の演算手段により前記命令が実行される都度、前記メモリに保持された構成情報にもとづいて値を更新することを特徴とする付記 6 に記載の情報処理装置。

【 0 0 9 0 】

（付記 8）前記所定の命令セット内の命令には、前記メモリに前記構成情報をロードする命令が含まれることを特徴とする付記 5 ～ 7 のいずれか一つに記載の情報処理装置。

【 0 0 9 1 】

（付記 9）さらに、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命



令であるか否かを判定する第 2 の判定手段と、

前記第 2 の判定手段により、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリに前記構成情報をロードする命令を複数発行する命令発行手段と、を備え、

前記第 1 の演算手段は、前記命令発行手段により発行された命令を実行することを特徴とする付記 5 ～ 7 のいずれか一つに記載の情報処理装置。

【 0 0 9 2 】

(付記 1 0) さらに、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定する第 2 の判定手段と、

前記第 2 の判定手段により、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリが割り当てられた所定のレジスタに前記構成情報を転送する命令を発行する命令発行手段と、を備え、

前記第 1 の演算手段は、前記命令発行手段により発行された命令を実行することを特徴とする付記 5 ～ 7 のいずれか一つに記載の情報処理装置。

【 0 0 9 3 】

(付記 1 1) さらに、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定する第 2 の判定手段と、

前記第 2 の判定手段により、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリが割り当てられたアドレス空間内の所定の領域に前記構成情報をストアする命令を発行する命令発行手段と、を備え、

前記第 1 の演算手段は、前記命令発行手段により発行された命令を実行することを特徴とする付記 5 ～ 7 のいずれか一つに記載の情報処理装置。

【 0 0 9 4 】

(付記 1 2) さらに、前記判定手段により前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定する第 2 の判定手段と、

前記第 2 の判定手段により、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリが割り当てられたアドレス空間内の所定の領域に前記構成情報を転送するよう DMA コントローラに指示する指示手段と、

を備えたことを特徴とする付記 5 ～ 7 のいずれか一つに記載の情報処理装置。

【 0 0 9 5 】

(付記 1 3) 前記第 2 の演算手段は、前記構成情報出力手段から出力された構成情報により決定される回路構成で、与えられたデータ中の任意のビットを入れ替えることを特徴とする付記 1 ～ 1 2 のいずれか一つに記載の情報処理装置。

【 0 0 9 6 】

(付記 1 4) 前記第 2 の演算手段は、前記構成情報出力手段から出力された構成情報により決定される回路構成で、与えられたデータ中の 1 の個数を計数することを特徴とする付記 1 ～ 1 2 のいずれか一つに記載の情報処理装置。

【 0 0 9 7 】

(付記 1 5) さらに、前記構成情報出力手段により出力された構成情報中所定のビット位置にある情報のみを出力する選択手段を備えたことを特徴とする付記 1 ～ 1 4 のいずれか一つに記載の情報処理装置。

【 0 0 9 8 】

(付記 1 6) 与えられた命令が所定の命令セット内の命令であるか否かを判定する判定工程と、

前記判定工程で前記命令が所定の命令セット内の命令であると判定された場合に、当該命令を実行する第 1 の演算工程と、

前記判定工程で前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令を実行するための回路構成を決定する構成情報を出力する構成情報出力工程と、

前記構成情報出力工程で出力された構成情報により決定される回路構成で前記命令を実行する第 2 の演算工程と、

を含んだことを特徴とする情報処理方法。

【 0 0 9 9 】

（付記 1 7）前記構成情報出力工程では、前記判定工程で前記命令が所定の命令セット内の命令でないと判定された場合に、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする付記 1 6 に記載の情報処理方法。

【 0 1 0 0 】

（付記 1 8）前記構成情報出力工程では、前記命令で指定されたアドレスおよび／または所定のレジスタに保持されたアドレスにもとづいて、複数の構成情報の中から当該命令を実行するための回路構成を決定する構成情報を選択して出力することを特徴とする付記 1 7 に記載の情報処理方法。

【 0 1 0 1 】

（付記 1 9）前記構成情報は書き換え可能なメモリに保持されることを特徴とする付記 1 6 ～ 1 8 のいずれか一つに記載の情報処理方法。

【 0 1 0 2 】

（付記 2 0）前記構成情報は前記メモリのほか、前記命令内の所定のフィールドあるいは所定のレジスタにも保持されることを特徴とする付記 1 9 に記載の情報処理方法。

【 0 1 0 3 】

（付記 2 1）前記所定のレジスタは、前記第 2 の演算工程で前記命令が実行される都度、前記メモリに保持された構成情報にもとづいて値を更新することを特徴とする付記 2 0 に記載の情報処理方法。

【 0 1 0 4 】

（付記 2 2）さらに、前記判定工程で前記命令が所定の命令セット内の命令でないと判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定する第 2 の判定工程と、

前記第 2 の判定工程で、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリに前記構成情報をロードする命令を複数発行する命令発行工程と、を含み、

前記第 1 の演算工程では、前記命令発行工程で発行された命令を実行することを特徴とする付記 1 9 ～ 2 1 のいずれか一つに記載の情報処理方法。

【 0 1 0 5 】

(付記 2 3) さらに、前記判定工程で前記命令が所定の命令セット内の命令でないとして判定された場合に、当該命令が前記メモリに前記構成情報をロードする命令であるか否かを判定する第 2 の判定工程と、

前記第 2 の判定工程で、前記命令が前記メモリに前記構成情報をロードする命令であると判定された場合に、前記メモリが割り当てられた所定のレジスタに前記構成情報を転送する命令を発行する命令発行工程と、を含み、

前記第 1 の演算工程では、前記命令発行工程で発行された命令を実行することを特徴とする付記 1 9 ～ 2 1 のいずれか一つに記載の情報処理方法。

【 0 1 0 6 】

(付記 2 4) さらに、前記構成情報出力工程で出力された構成情報中所定のビット位置にある情報のみを出力する選択工程を含んだことを特徴とする付記 1 6 ～ 2 3 のいずれか一つに記載の情報処理方法。

【 0 1 0 7 】

【発明の効果】

以上説明したように本発明により実現されるプロセッサは、与えられた命令が命令セット内に用意されていないような命令（たとえば入力データ中のビットを入れ替える処理や、1 の個数を係数する処理など）であっても、メモリから読み出した構成情報により自らの演算器の構成を変化させることで、高速に当該命令を実行できるので、個々に回路を作り替えたり大規模な回路を搭載したりしなくても、ユーザが定義する様々な命令を高速に処理することが可能な情報処理装置が得られるという効果を奏する。

【図面の簡単な説明】

【図 1】

従来技術によるプロセッサの内部構成を模式的に示すブロック図である。

【図 2】

本発明によるプロセッサの内部構成を模式的に示す説明図である。

【図 3】

本発明によるコンフィギュレーションメモリ 2 0 6 の内容を模式的に示す説明

図である。

【図 4】

本発明によるプロセッサに与えられる命令コードの、フォーマットの一例を模式的に示す説明図である。

【図 5】

本発明の実施の形態 1 による第 2 演算器 2 0 5 の内部構成を模式的に示す説明図である。

【図 6】

本発明の実施の形態 2 による第 2 演算器 2 0 5 の内部構成を模式的に示す説明図である。

【図 7】

本発明の実施の形態 2 による第 2 演算器 2 0 5 で実行される、入力データ中の 1 の個数の計数処理の手順を示すフローチャートである。

【図 8】

図 7 に示すフローの 1 回目の処理を具体例により示す説明図である。

【図 9】

図 7 に示すフローの 2 回目の処理を具体例により示す説明図である。

【図 1 0】

図 7 に示すフローの 3 回目の処理を具体例により示す説明図である。

【図 1 1】

本発明の実施の形態 2 による第 2 演算器 2 0 5 でのシフト量およびマスクパターンを示す説明図である。

【図 1 2】

自動更新機能を具備する特定用途レジスタの回路構成の一例を示す説明図である。

【図 1 3】

本発明の実施の形態 3 による第 2 演算器 2 0 5 の内部構成を模式的に示す説明図である。

【図 1 4】

本発明の実施の形態 3 による第 2 演算器 2 0 5 で実行される、入力データの I P 転置処理の手順を示すフローチャートである。

【図 1 5】

I P 転置前のデータの一例を示す説明図である。

【図 1 6】

I P 転置後のデータの一例を示す説明図である。

【図 1 7】

本発明の実施の形態 3 による第 2 演算器 2 0 5 内の、ビット入れ替え器 1 3 0 6 におけるビットの入れ替え規則を示す説明図である。

【図 1 8】

図 1 4 に示すステップ S 1 4 0 1 の処理を具体例により示す説明図である。

【図 1 9】

図 1 4 に示すステップ S 1 4 0 2 の処理を具体例により示す説明図である。

【図 2 0】

図 1 4 に示すステップ S 1 4 0 5 の処理を具体例により示す説明図である。

【図 2 1】

図 1 4 に示すステップ S 1 4 0 6 の処理を具体例により示す説明図である。

【図 2 2】

コンフィギュレーションメモリ 2 0 6 をテーブルメモリとして使用する場合は、第 2 演算器 2 0 5 の回路構成の一例を示す説明図である。

【図 2 3】

コンフィギュレーションメモリ 2 0 6 をテーブルメモリとして使用する場合は、第 2 演算器 2 0 5 の回路構成の他の一例を示す説明図である。

【図 2 4】

コンフィギュレーションメモリ 2 0 6 をテーブルメモリとして使用する場合は、第 2 演算器 2 0 5 の回路構成の他の一例を示す説明図である。

【符号の説明】

1 0 0, 2 0 0 命令キャッシュ

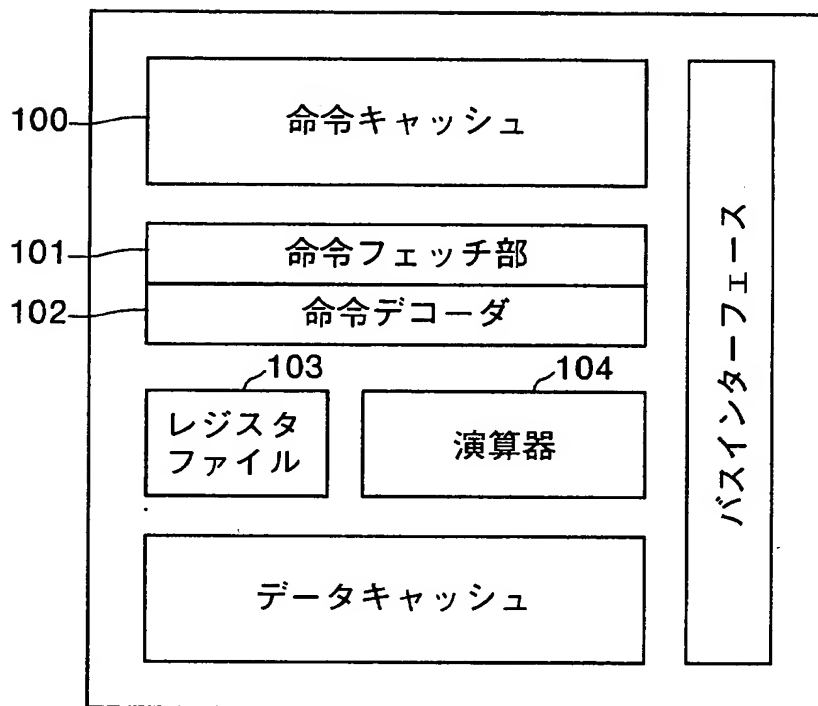
1 0 1, 2 0 1 命令フェッチ部

1 0 2, 2 0 2 命令デコーダ
1 0 3, 2 0 3 レジスタファイル
1 0 4, 2 0 4 演算器
2 0 5 第2演算器
2 0 6 コンフィギュレーションメモリ
6 0 0, 1 3 0 0 右シフタ
6 0 1, 1 3 0 1 パターンジェネレータ
6 0 2, 6 0 3, 1 3 0 3, 1 3 0 4 AND回路
6 0 4 加算器
1 3 0 2 反転回路
1 3 0 5 OR回路
1 3 0 6 ビット入れ替え器

【書類名】 図面

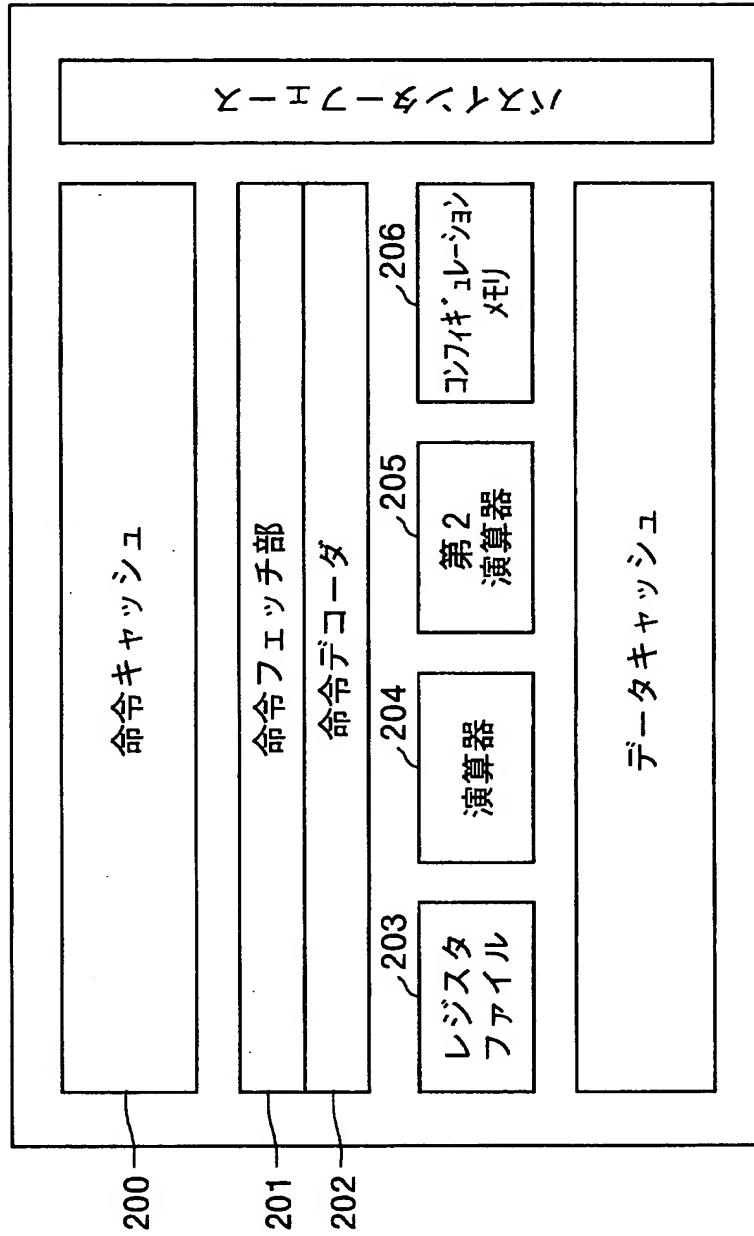
【図 1】

従来技術によるプロセッサの内部構成を模式的に示すブロック図



【図 2】

本発明によるプロセッサの内部構成を模式的に示す説明図



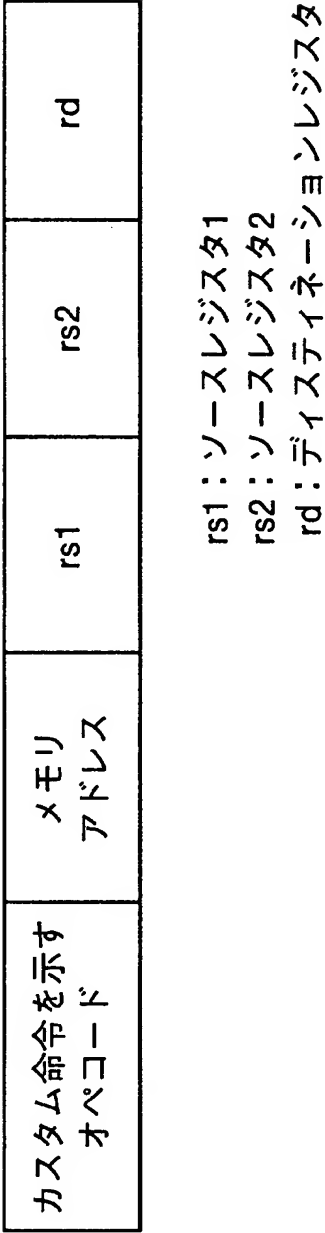
【図 3】

本発明によるコンフィギュレーションメモリ 2 0 6 の内容を
模式的に示す説明図

第 1 のカスタム命令の構成情報
第 2 のカスタム命令の構成情報
第 3 のカスタム命令の構成情報
・ ・ ・ ・
第nのカスタム命令の構成情報

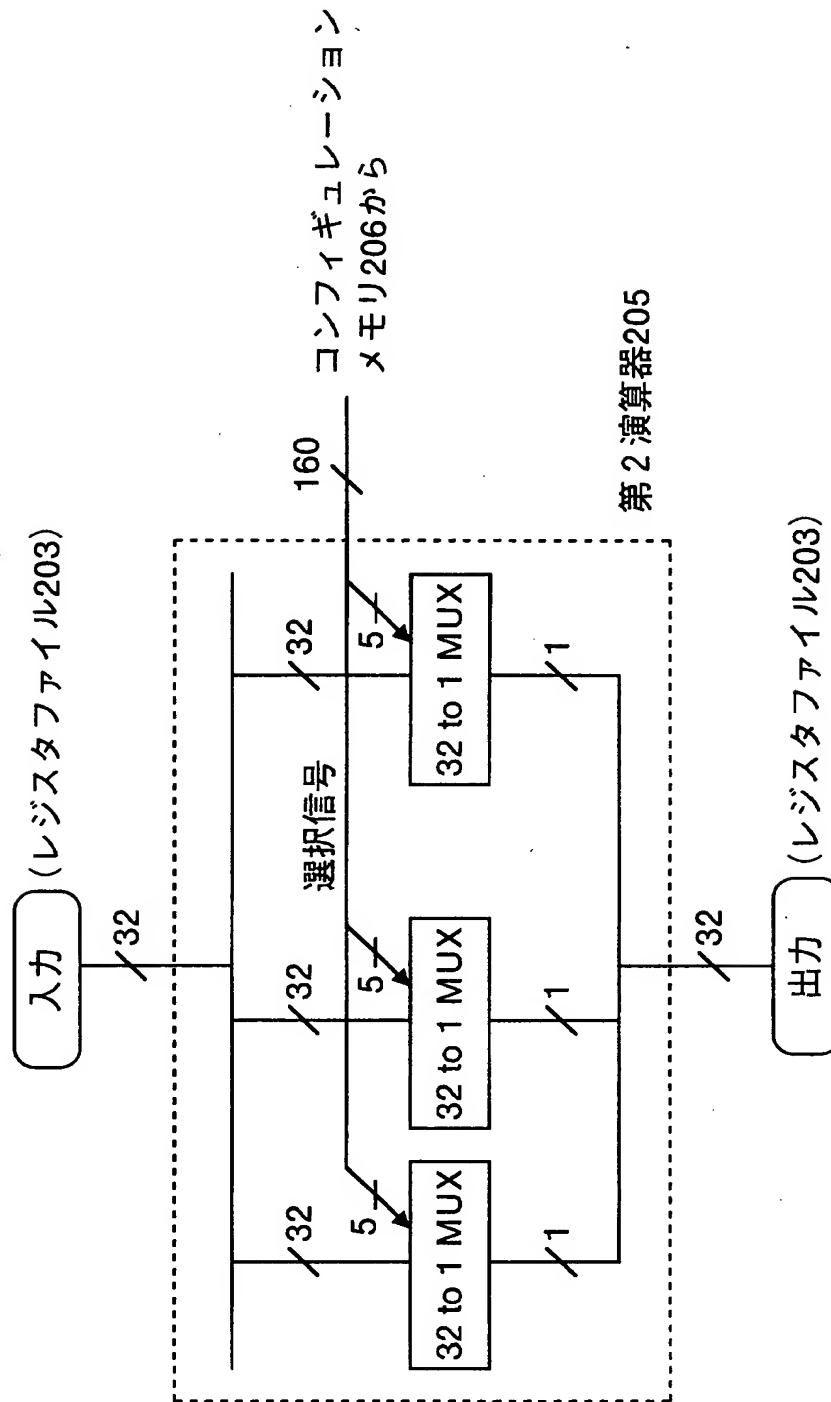
【図 4】

本発明によるプロセッサに与えられる命令コードの、フォーマットの一例を模式的に示す説明図



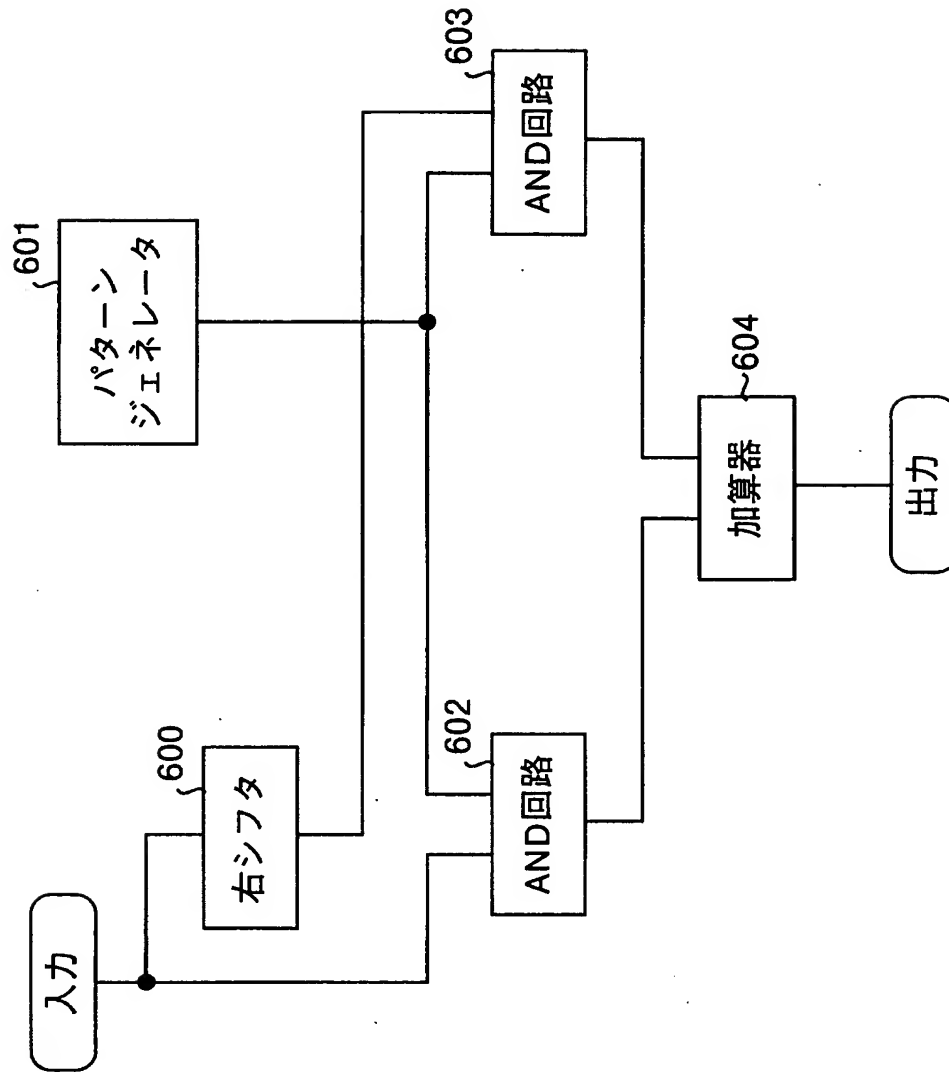
【図 5】

本発明の実施の形態 1 による第 2 演算器 205 の内部構成を模式的に示す説明図



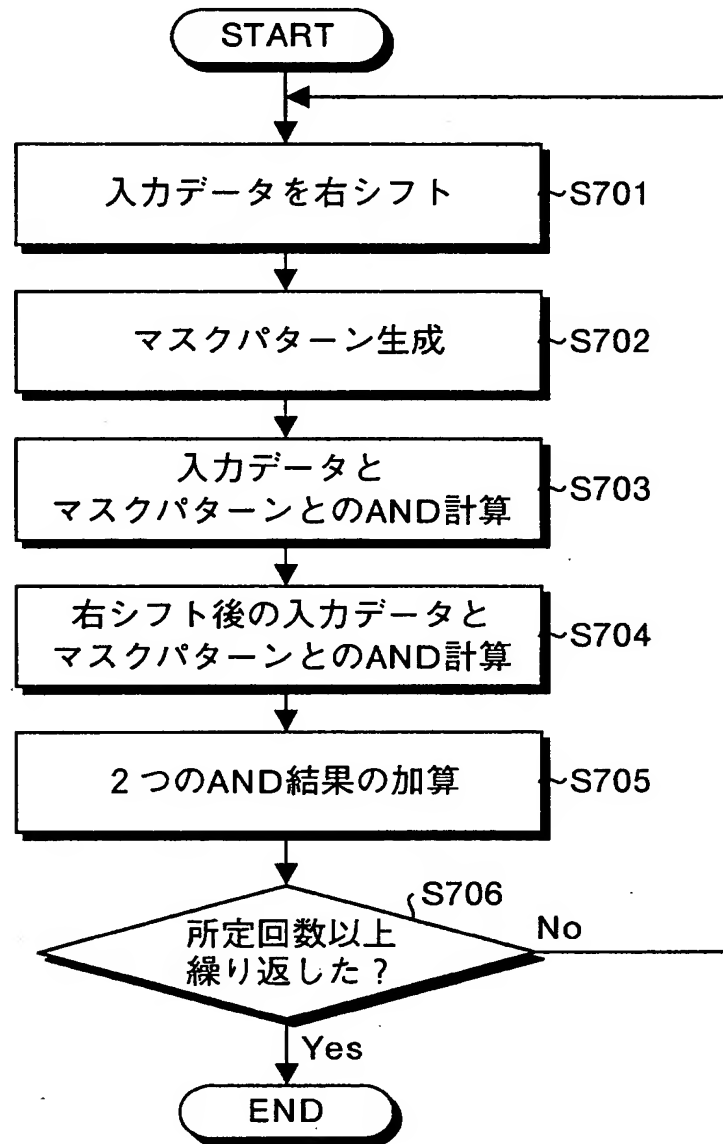
【図 6】

本発明の実施の形態 2 による第 2 演算器 2 0 5 の内部構成を模式的に示す説明図



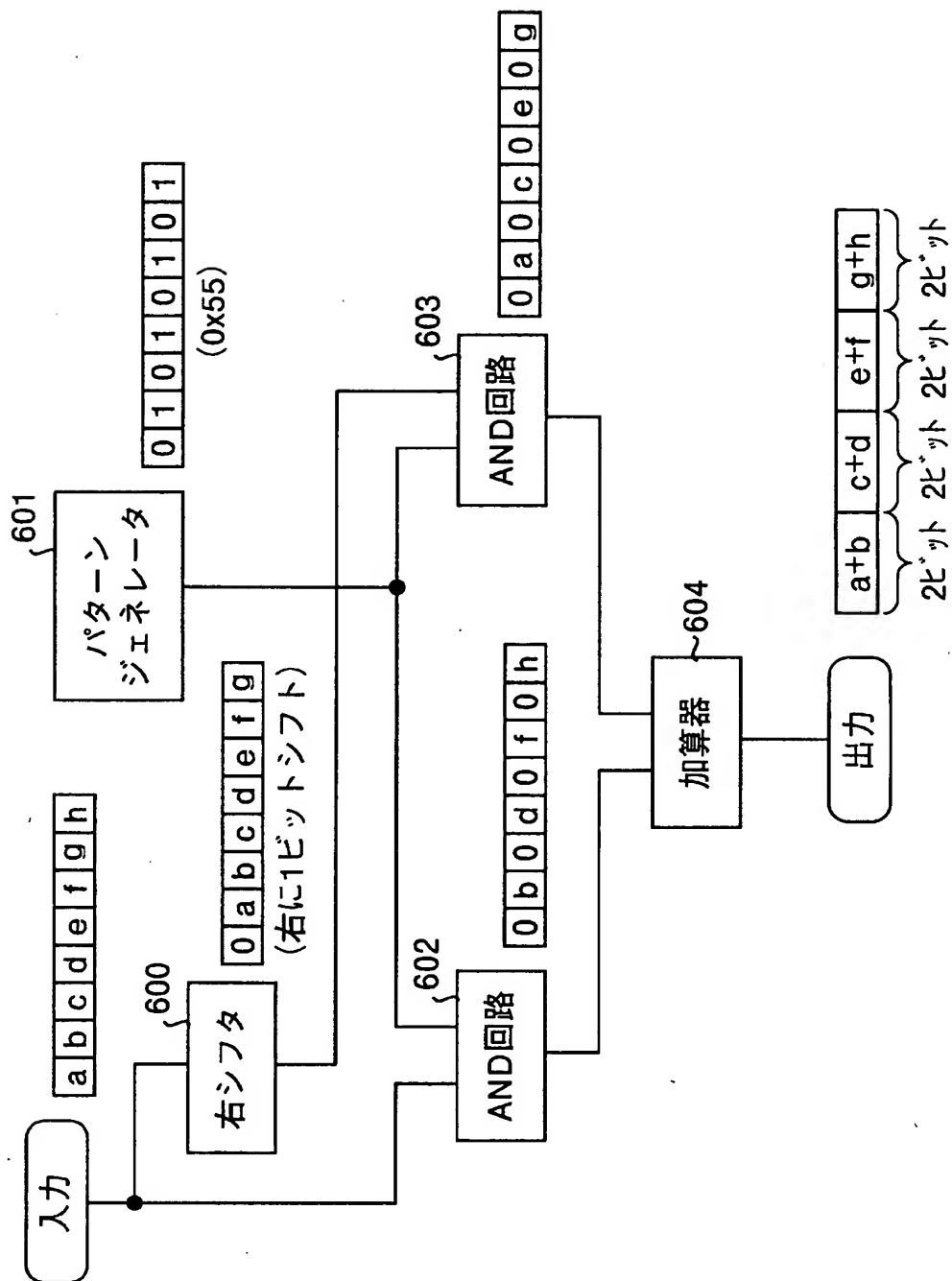
【図 7】

本発明の実施の形態 2 による第 2 演算器 2 0 5 で実行される、
入力データ中の 1 の個数の計数処理の手順を示すフローチャート



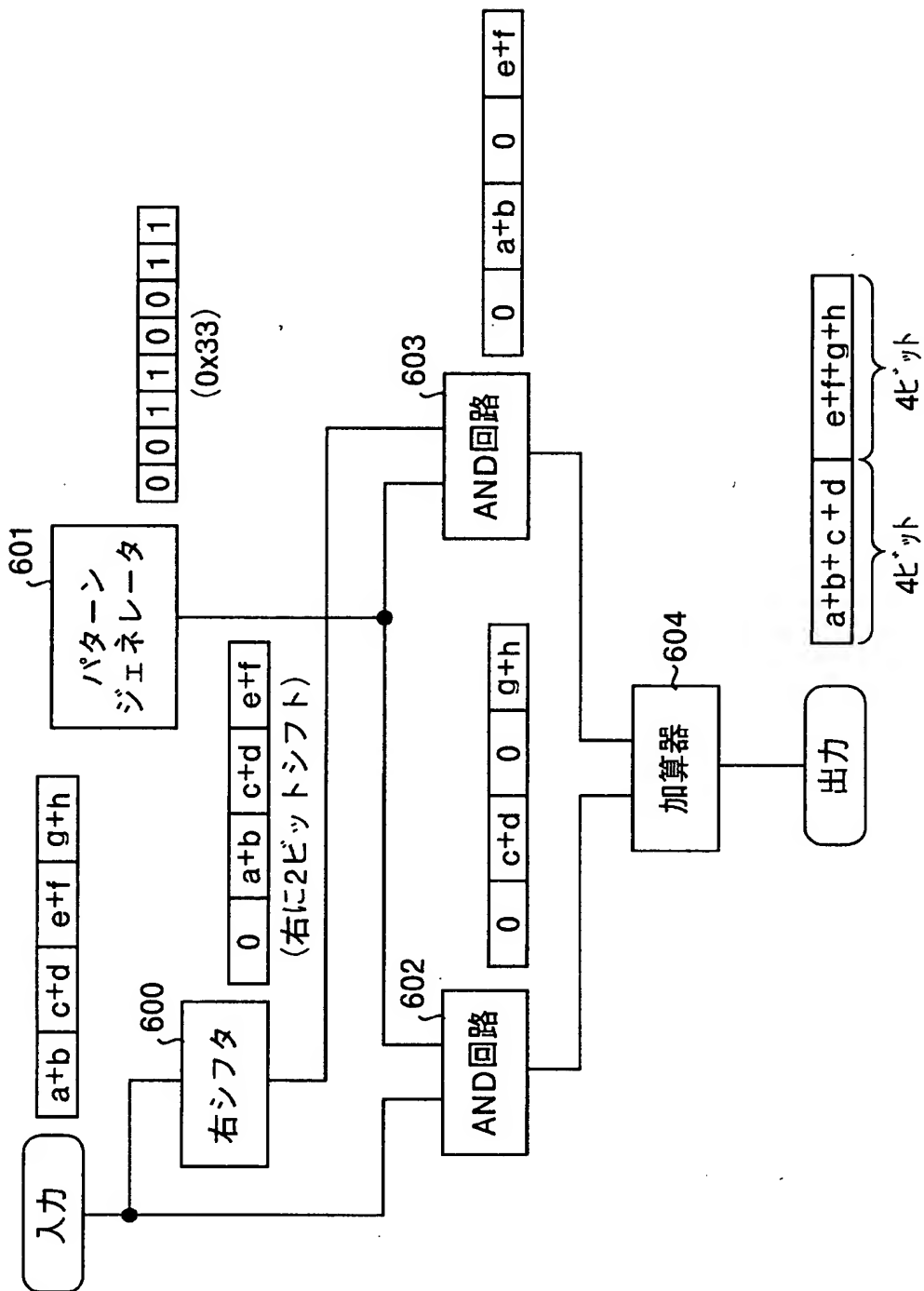
【図 8】

図 7 に示すフローの 1 回目の処理を具体例により示す説明図



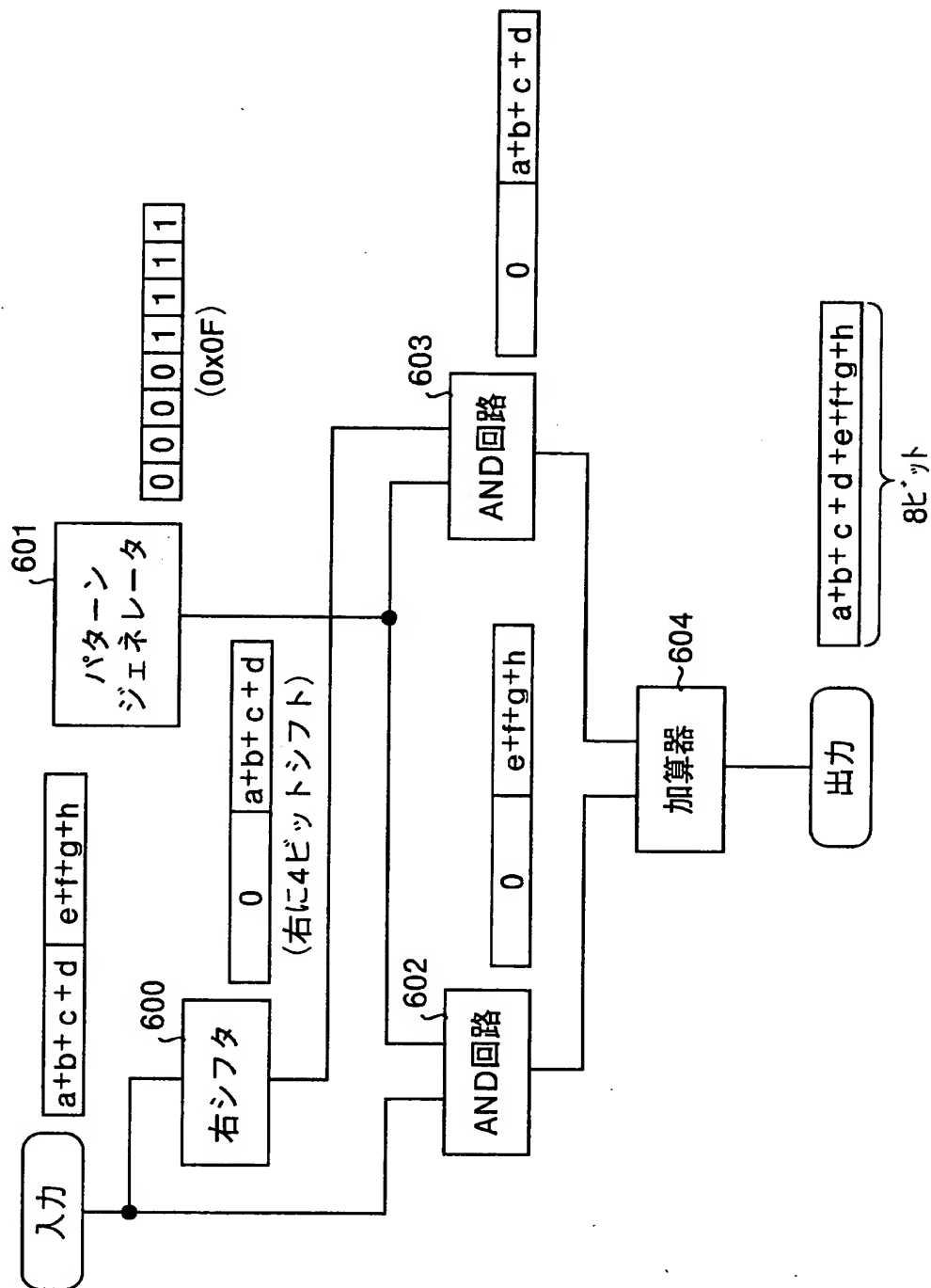
【図 9】

図 7 に示すフローの 2 回目の処理を具体例により示す説明図



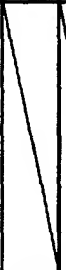
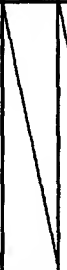
【図 1 0】

図 7 に示すフローの 3 回目の処理を具体例により示す説明図



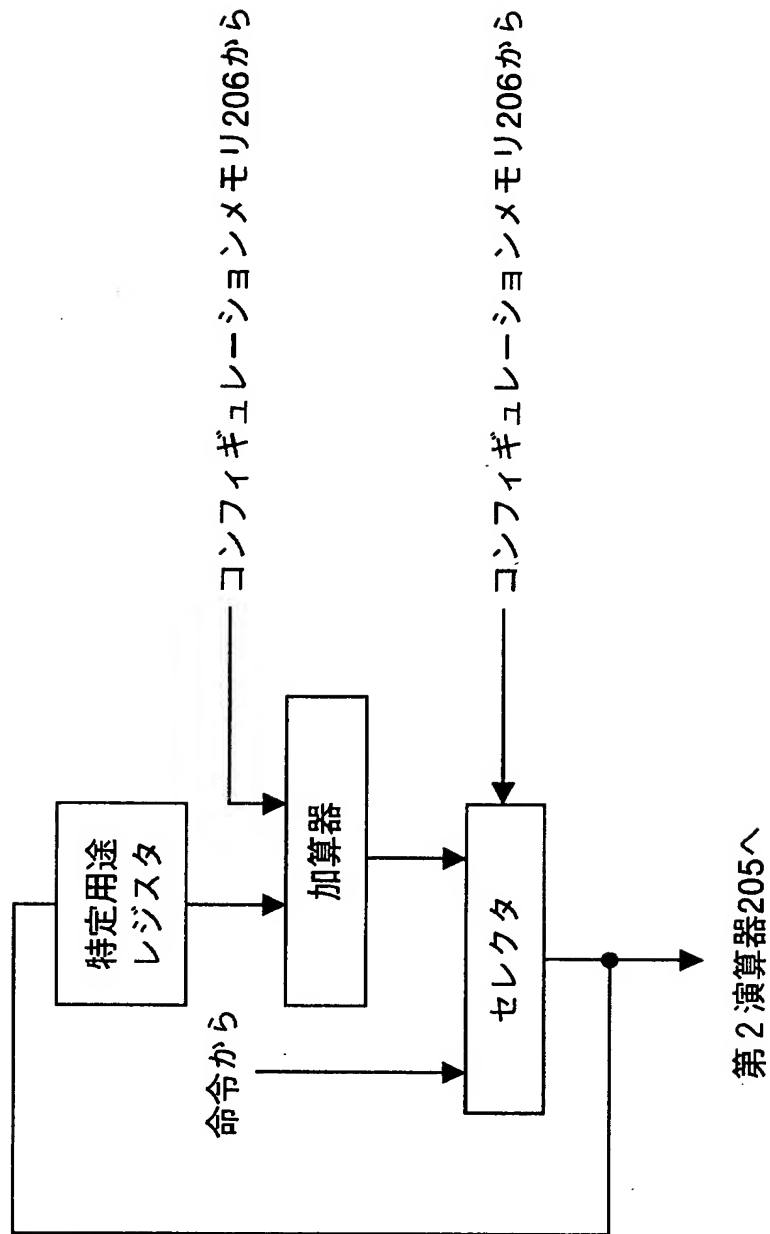
【図 1 1】

本発明の実施の形態 2 による第 2 演算器 2 0 5 でのシフト量およびマスクパターンを示す説明図

		8 ビット	32 ビット
1 回目	シフト量	1	1
	マスクパターン	01010101(0x55)	01010101 01010101 01010101 01010101 (0x55555555)
2 回目	シフト量	2	2
	マスクパターン	00110011(0x33)	00110011 00110011 00110011 00110011 (0x33333333)
3 回目	シフト量	4	4
	マスクパターン	00001111(0x0F)	00001111 00001111 00001111 00001111 (0x0F0F0F0F)
4 回目	シフト量		
	マスクパターン		
5 回目	シフト量		
	マスクパターン		

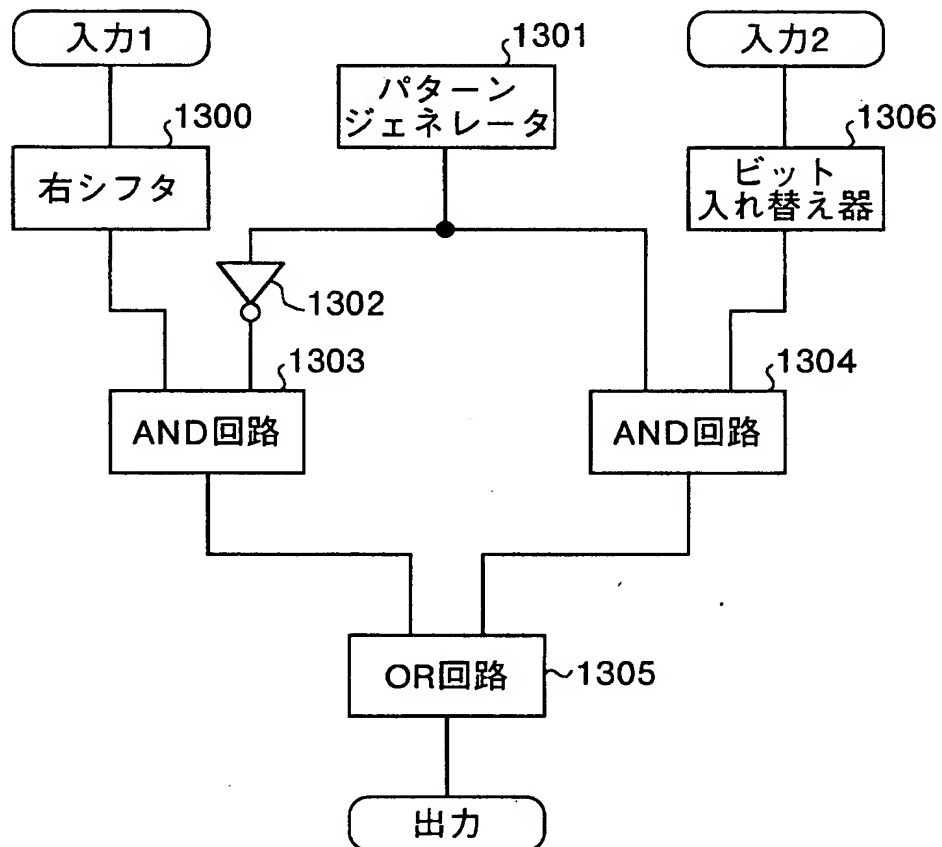
【図 1 2】

自動更新機能を具備する特定用途レジスタの回路構成の一例を示す説明図



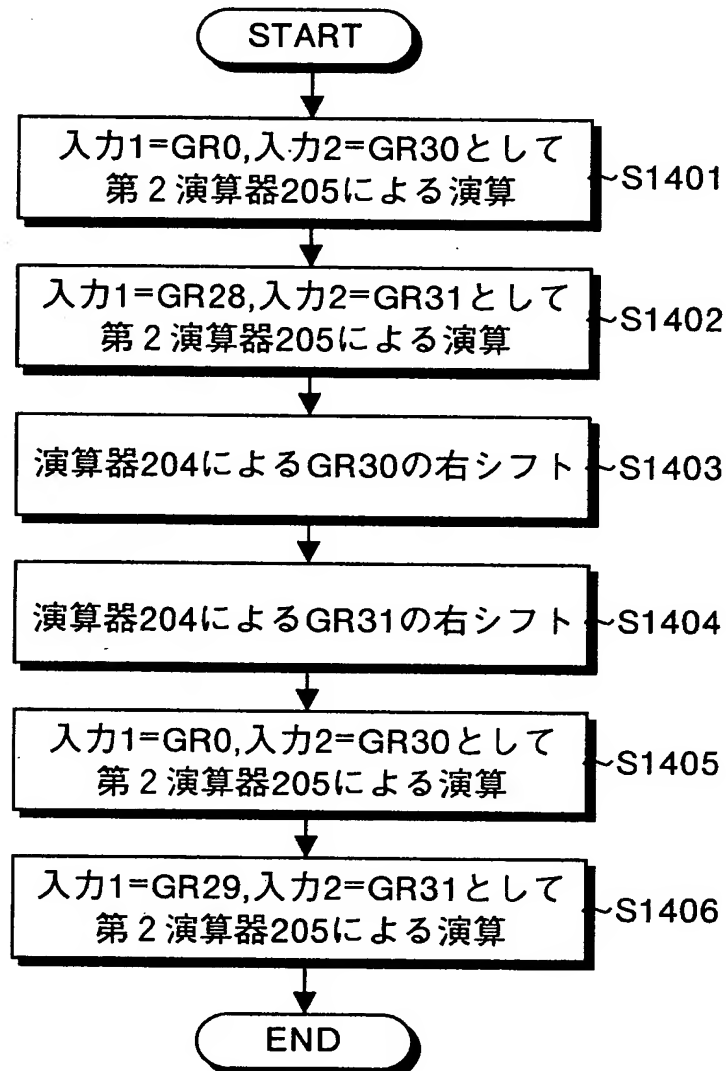
【図 1 3】

本発明の実施の形態 3 による第 2 演算器 2 0 5 の内部構成を模式的に示す説明図



【図 1 4】

本発明の実施の形態 3 による第 2 演算器 2 0 5 で実行される、
入力データの I P 転置処理の手順を示すフローチャート



【図 1 5】

Ⅰ P 転置前のデータの一例を示す説明図

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

【図 1 6】

Ⅰ P 転置後のデータの一例を示す説明図

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

【図 1 7】

本発明の実施の形態 3 による第 2 演算器 2 0 5 内の、
ビット入れ替え器 1 3 0 6 におけるビットの入れ替え規則を示す説明図

(a) ビット入れ替え前

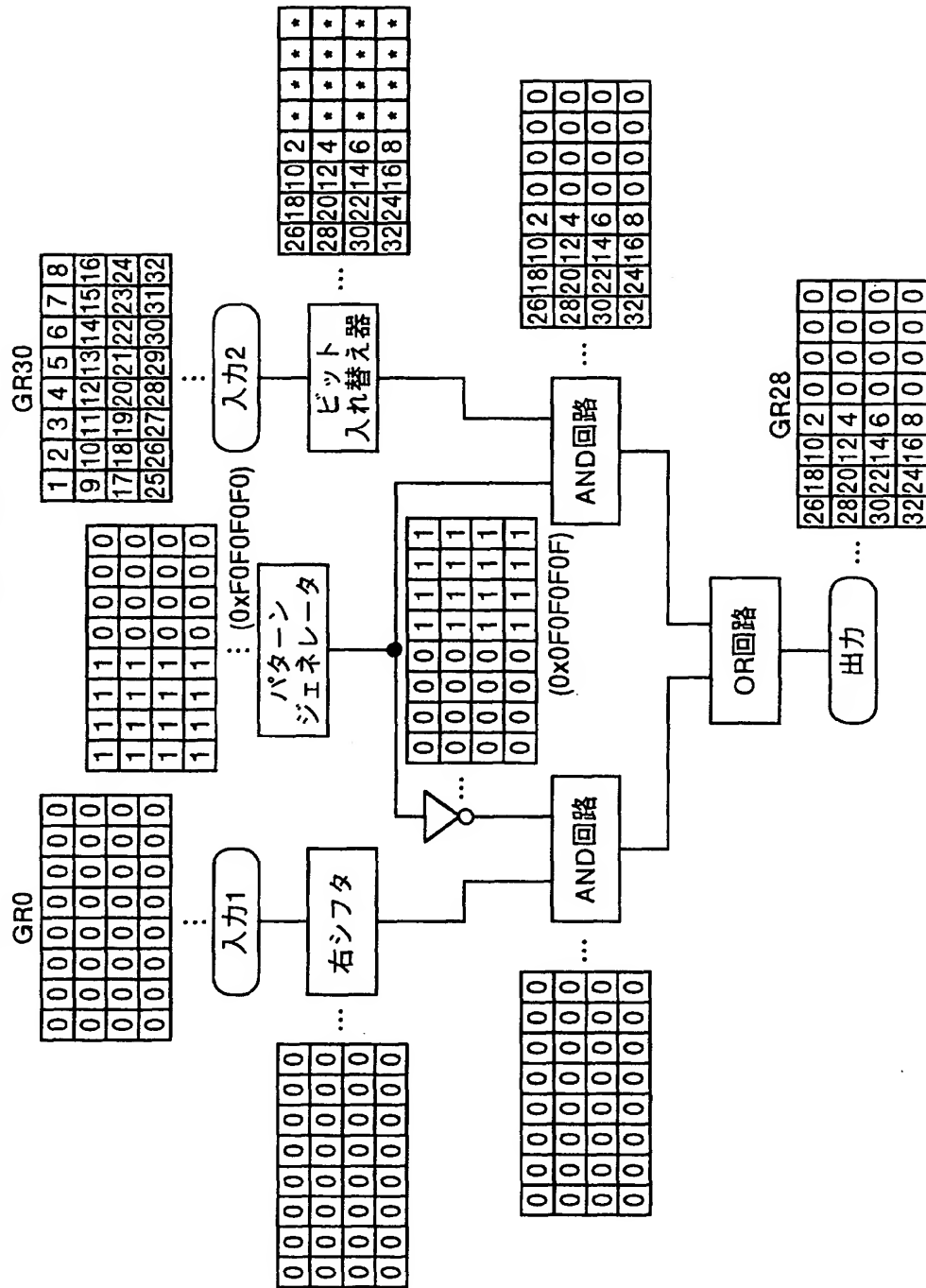
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

(b) ビット入れ替え後

26	18	10	2	*	*	*	*	28	20	12	4	*	*	*	*	30	22	14	6	*	*	*	*	32	24	16	8	*	*	*	*
----	----	----	---	---	---	---	---	----	----	----	---	---	---	---	---	----	----	----	---	---	---	---	---	----	----	----	---	---	---	---	---

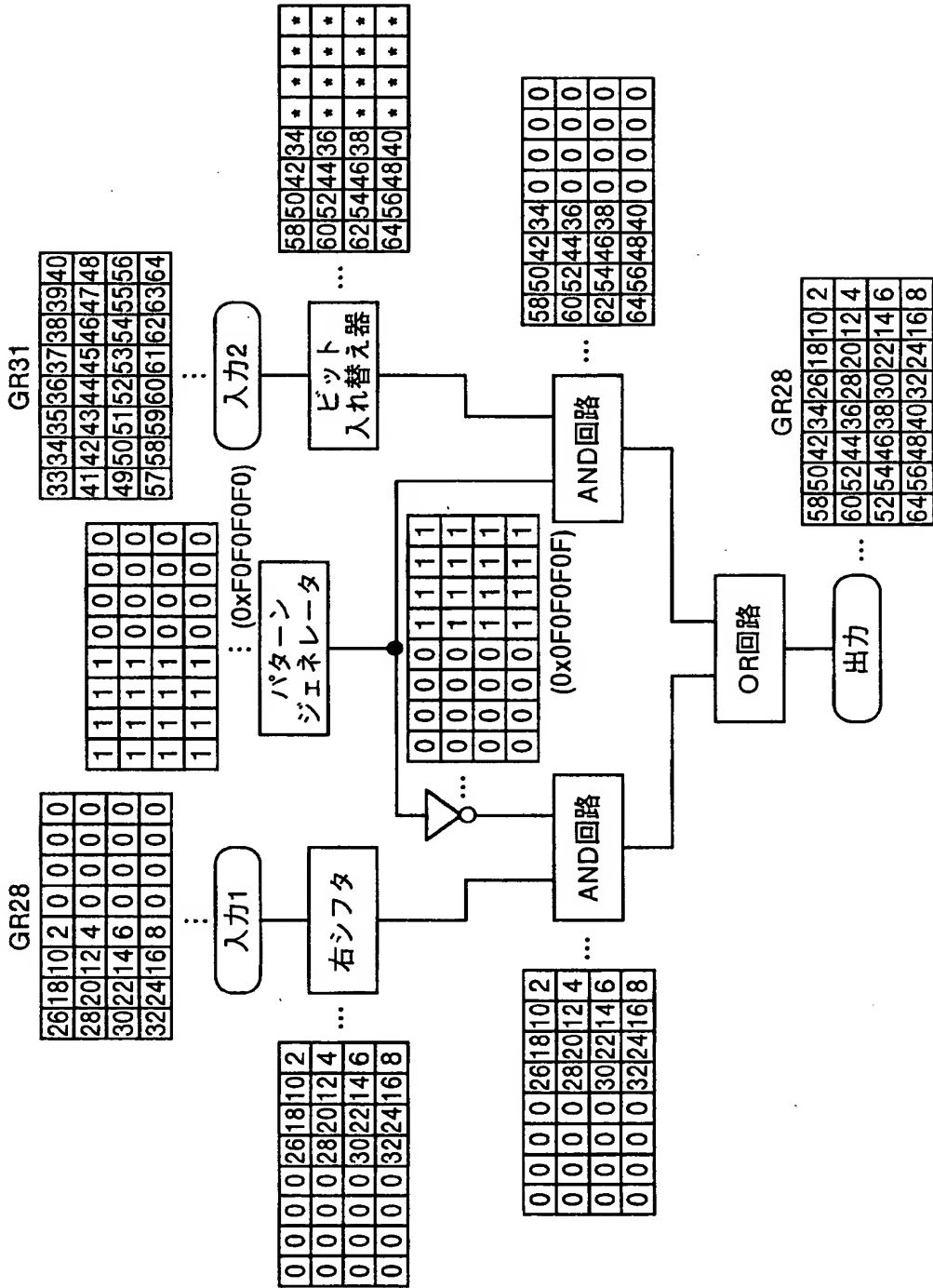
【図 1 8】

図 1 4 に示すステップ S 1 4 0 1 の処理を具体例により示す説明図



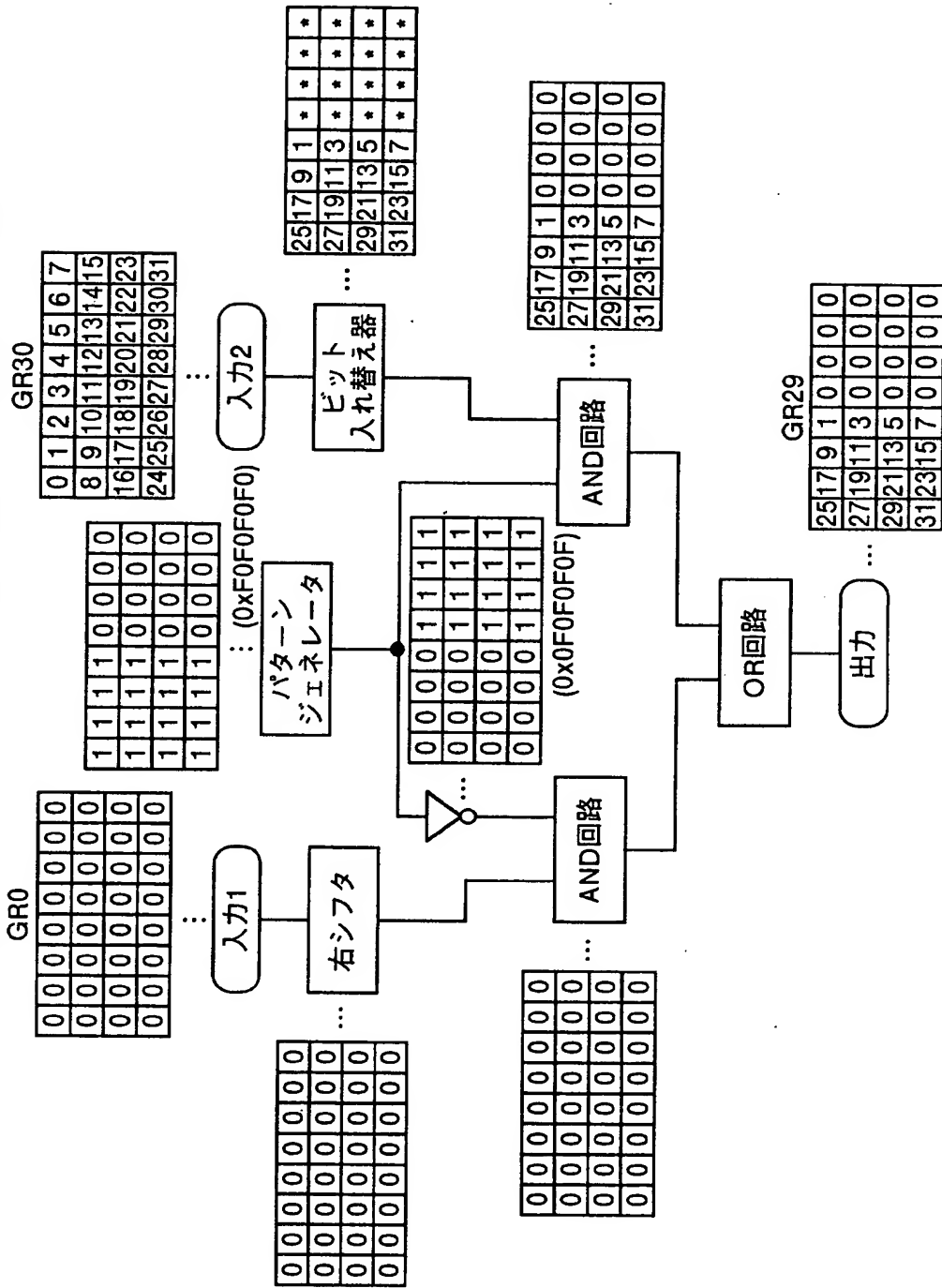
【図 19】

図 14 に示すステップ S1402 の処理を具体例により示す説明図



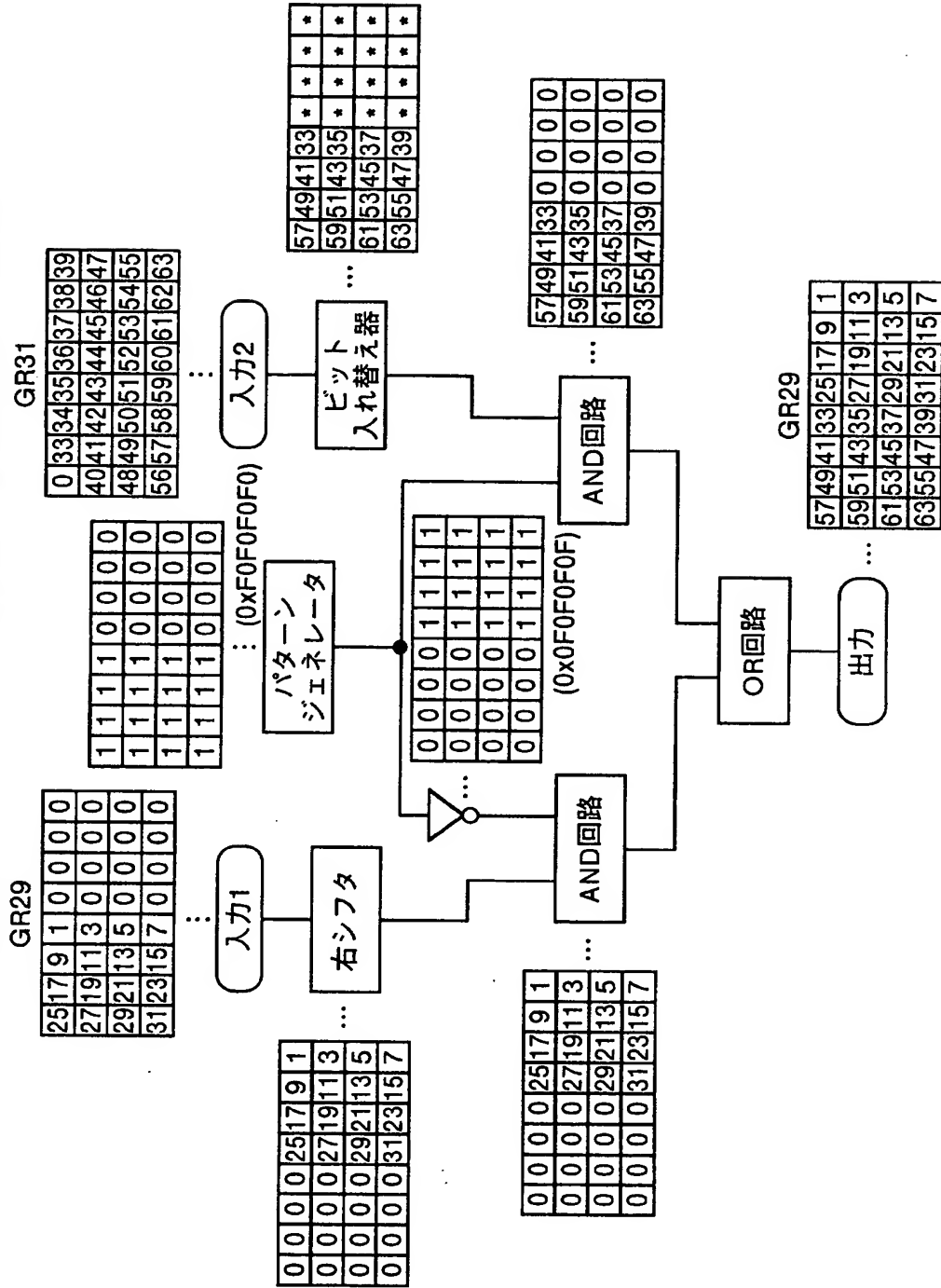
【図 2 0】

図 1 4 に示すステップ S 1 4 0 5 の処理を具体例により示す説明図



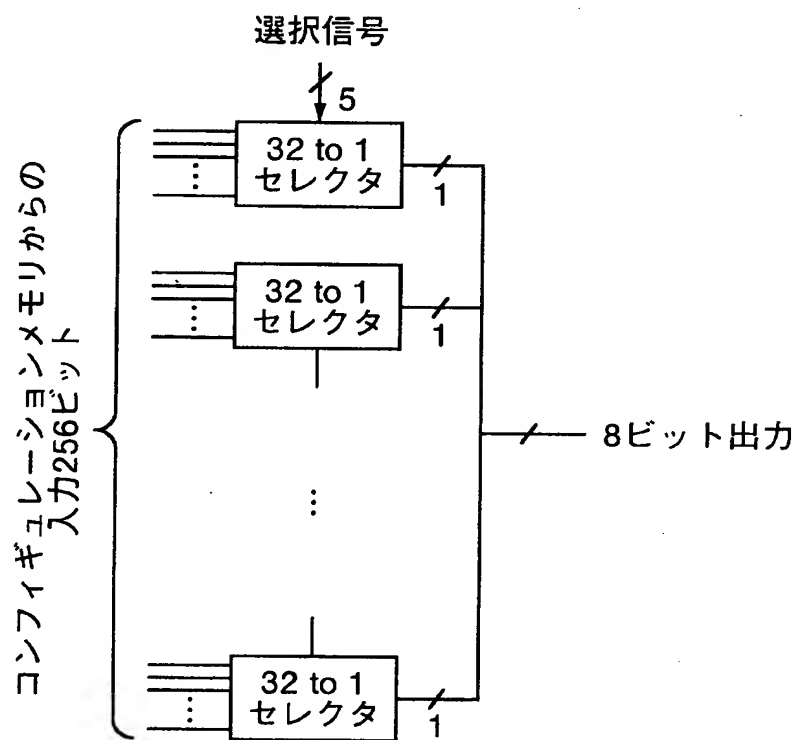
【図 2 1】

図 1 4 に示すステップ S 1 4 0 6 の処理を具体例により示す説明図



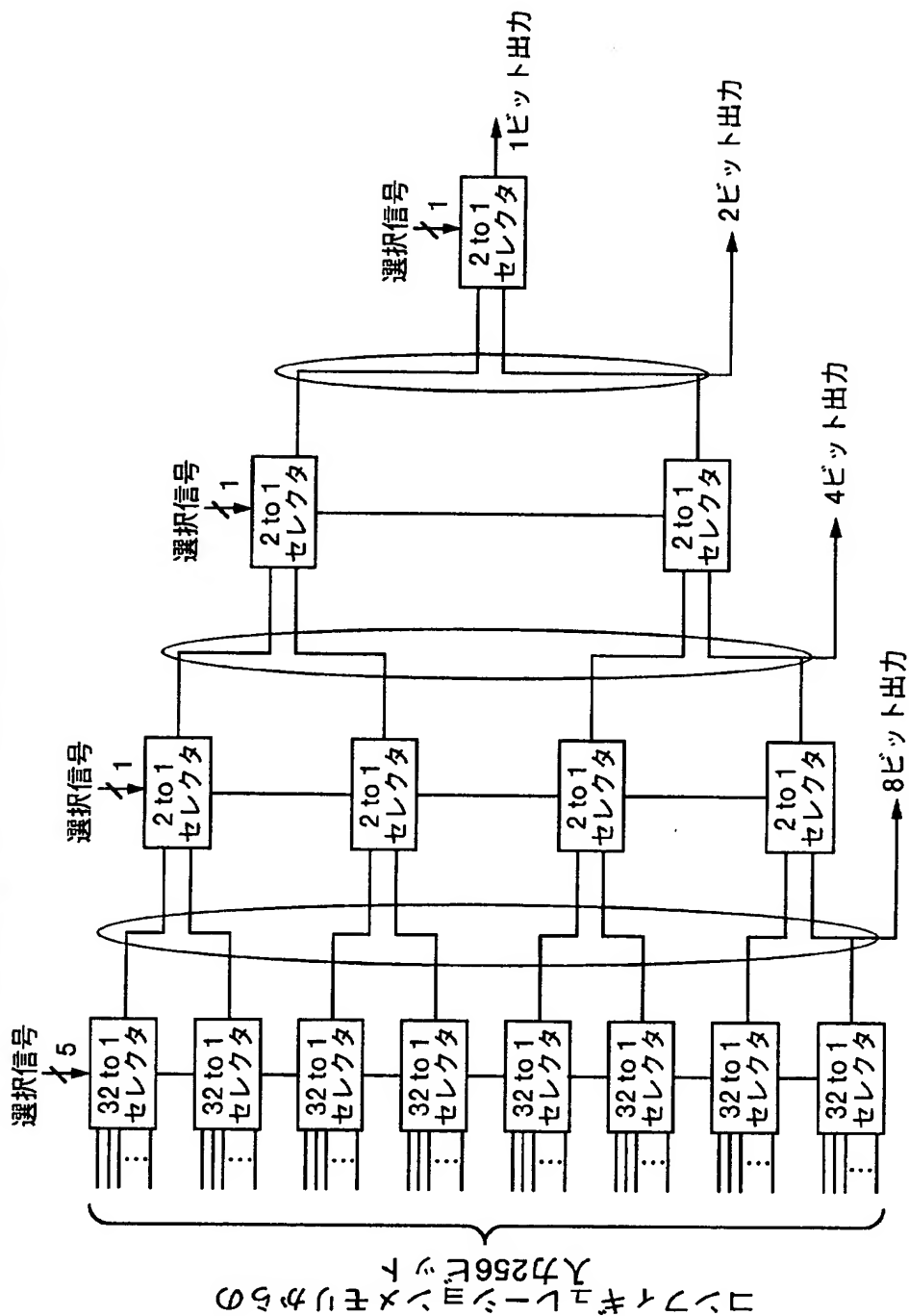
【図 2 2】

コンフィギュレーションメモリ 2 0 6 をテーブルメモリとして
 使用する場合の、第 2 演算器 2 0 5 の回路構成の一例を示す説明図



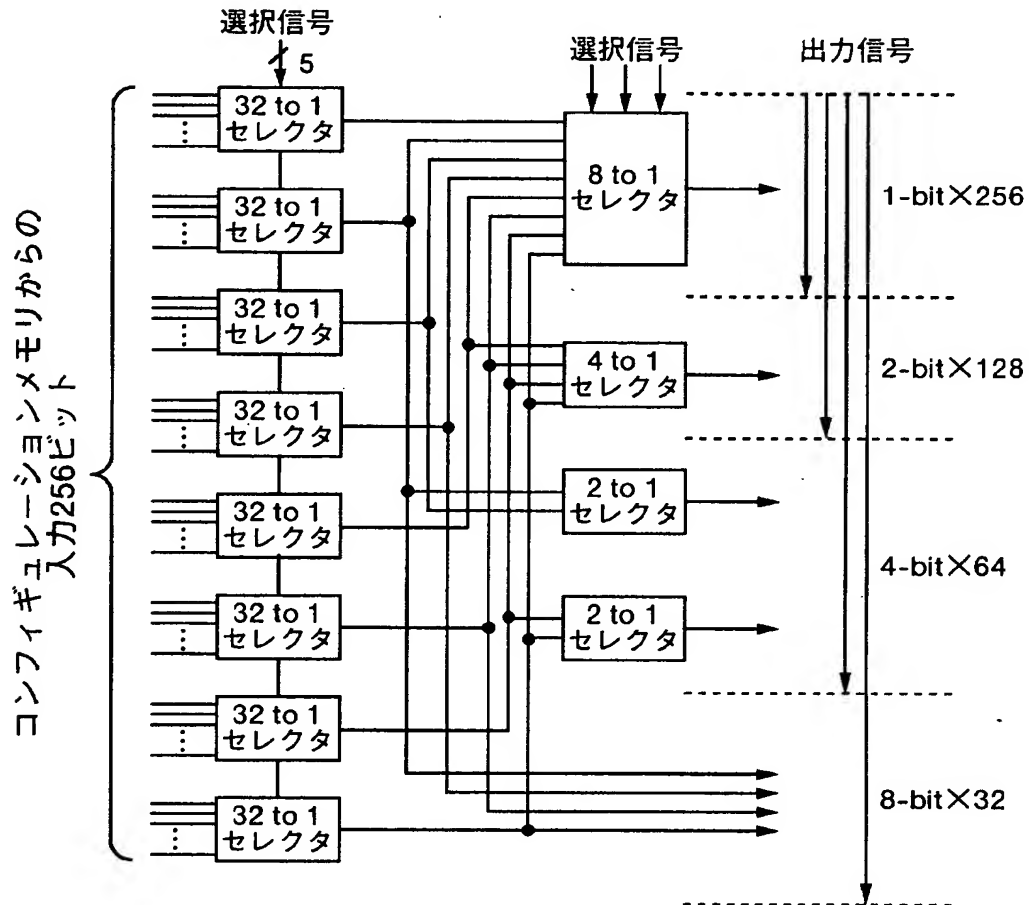
【図 23】

コンフィギュレーションメモリ206をテーブルメモリとして使用する場合の、
第2演算器205の回路構成の他の一例を示す説明図



【図 2 4】

コンフィギュレーションメモリ 2 0 6 をテーブルメモリとして
 使用する場合の、第 2 演算器 2 0 5 の回路構成の他の一例を示す説明図



【書類名】 要約書

【要約】

【課題】 個々にプロセッサを作り替えたり、プロセッサの外部に大規模な回路を搭載したりしなくても、ユーザが定義する様々な命令を高速に処理することが可能なプロセッサを提供すること。

【解決手段】 本発明によるプロセッサは、命令セット内の命令は演算器 2 0 4 で、それ以外のカスタム命令は第 2 演算器 2 0 5 でそれぞれ実行する。第 2 演算器 2 0 5 は具体的には、複数の AND 回路、OR 回路、加算器、セレクタ、マルチプレクサなどから構成され、どの命令が入力したときにどの回路とどの回路とを組み合わせればよいかは、コンフィギュレーションメモリ 2 0 6 に構成情報として保持されている。1 つのカスタム命令に 1 つの構成情報が対応しており、当該情報により決定される最適な回路構成の第 2 演算器 2 0 5 で当該命令を実行することで、演算器 2 0 4 で実行する場合よりも処理の高速化が図れる。

【選択図】 図 2

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日
[変更理由] 住所変更
住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社